Titre: Title:	Multi-Unit Optimization for a System with Mutiple Non-Identical Units and Multipe Inputs-Application to Photovoltaic Arrays
Auteur: Author:	Samin Sadre Bazaz
Date:	2013
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Sadre Bazaz, S. (2013). Multi-Unit Optimization for a System with Mutiple Non-Identical Units and Multipe Inputs-Application to Photovoltaic Arrays [Master's thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/1156/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/1156/
Directeurs de recherche: Advisors:	Michel Perrier, & Lyne Woodward
Programme: Program:	Génie chimique

UNIVERSITÉ DE MONTRÉAL

MULTI-UNIT OPTIMIZATION FOR A SYSTEM WITH MULTIPLE NON-IDENTICAL UNITS AND MULTIPE INPUTS- APPLICATION TO PHOTOVOLTAIC ARRAYS

SAMIN SADRE BAZAZ

DÉPARTEMENT DE GÉNIE CHIMIQUE

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

(GÉNIE CHIMIQUE)

MAI 2013

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

MULTI-UNIT OPTIMIZATION FOR A SYSTEM WITH MULTIPLE NON-IDENTICAL UNITS AND MULTIPE INPUTS- APPLICATION TO PHOTOVOLTAIC ARRAYS

présenté par : SADRE BAZAZ Samin

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. HENRY Olivier, Ph.D., président

M. PERRIER Michel, Ph.D., membre et directeur de recherche

Mme WOODWARD Lyne, Ph.D., membre et codirectrice de recherche

M. GOURDEAU Richard, Ph.D., membre

DEDICATION

To Ghasem, Zari, Sahar, and Mohammad

for their endless love and support to me

ACKNOWLEDGEMENTS

I owe sincere thankfulness to my research supervisor, Professor Michel Perrier, for his understanding and encouragement. I am sure that this research would not have been possible without his help, academic and financial support, and patience.

I would also like to show my gratitude to my co-supervisor, Professor Lyne Woodward, who made me believe in myself and guided me step by step through many discussions. This thesis would not have been possible without her advice and knowledge of optimization.

Special thanks to my colleague and friends in Polytechnique Montreal, Farhad Azar for his advices in the first steps of this research, Masood Khaksar for the helpful discussions we had, and Tatiana Rafione for her time to help me writing the *Résumé en Francais*. I made many friends in Polytechnique Montreal and I would like to thank to my dear friends Shadi Sharif, Salomeh Chegini, Banafsheh Gilani, Moye Ajao, Hamed Bashiri, Behrang Mansoornejad, and Jean-christophe Bonhivers for their valuable friendship and inspiration.

My parents Zari and Ghasem, and my sister Sahar have given to me their unequivocal support throughout, as always, for which my mere expression of thanks likewise does not suffice. Last, but by no means least, I would like to thank my love Mohammad for his personal support, great patience, and smile at all times. I love you!

RÉSUMÉ

L’optimisation est devenue un domaine clé dans l’industrie de transformation pour rester compétitif sur le marché mondial, s’adapter aux nouvelles contraintes environnementales et supporter l’augmentation des coûts énergétiques. Pour répondre à ces nouvelles exigences, les industries se doivent d’optimiser leurs installations afin de réduire les coûts d'exploitation, améliorer l'efficacité de la production, répondre aux spécifications de qualité des produits et sécurité des procédés. Avec le développement de nouvelles technologies de contrôle, il est aujourd’hui possible de maintenir un procédé à son point d’opération optimal.

L’optimisation en temps réel (RTO) est un outil permettant d’amener et maintenir un système à son point de fonctionnement optimal. Ce domaine de recherche a reçu une attention considérable dans l'industrie des procédés. Les méthodes d’optimisation en temps réels permettent de contrôler le comportement d’un procédé en ajustant les points de consigne des régulateurs de procédé pour suivre les changements de conditions opératoires et les perturbations externes qui prennent place au sein d’une usine.

Parmi les différentes approches d’optimisation en temps réel, les méthodes de commande extrémale sont celles qui permettent de satisfaire les conditions nécessaires d'optimalité. Dans la commande extrémale, l'optimisation est traitée comme un problème de contrôle du gradient de la fonction objectif à zéro. La principale différence entre les diverses méthodes de commande extrémale repose sur la façon dont le gradient est estimé. La plupart de ces méthodes impliquent l’application d’une perturbation temporelle périodique. De plus, afin d’isoler les effets de la dynamique du système sur le gradient estimé, une séparation de plusieurs échelles de temps est requise.

La méthode d’optimisation multi-unités est une méthode de commande extrémale dans laquelle la perturbation est appliquée entre les unités plutôt que sur un domaine temporel. Une séparation d'échelle de temps n'est plus nécessaire. La convergence est de ce fait plus rapide.

La méthode d’optimisation multi-unités nécessite la présence de plusieurs unités identiques, chacune d'entre elles fonctionnant à des valeurs d'entrée qui diffèrent par une constante prédéterminée de décalage. Bien que cette méthode soit utile lorsque le système se compose de plusieurs unités, la convergence au point optimal a seulement été prouvée pour des unités au sein d’un procédé parfaitement identiques ou lorsqu’il y a seulement deux unités non identiques. En pratique, cette hypothèse est rarement vérifiée puisqu’un procédé industriel réel peut avoir plus de deux unités non identiques. Par conséquent, dans cette étude, une méthode d'optimisation basée sur l’optimisation multi-unités est proposée pour répondre à cette problématique. L'algorithme proposé est pour le cas d'une fonction objectif statique convexe avec deux entrées. L’algorithme comporte entre autre des corrections successives pour compenser les différences entre les surfaces statiques des fonctions objectif associées à chaque unité.

La dernière partie de cette thèse contient l'étude de cas où la méthode d'optimisation multi-unités est utilisée pour déterminer la puissance électrique maximale de panneaux photovoltaïques. L'électricité est principalement produite à partir de combustibles fossiles, de combustible nucléaire et de ressources renouvelables telles que le soleil, le vent, l'eau et la biomasse. L'énergie solaire est de plus en plus considérée pour la production de bioénergie et ce, en raison des récents progrès dans la fabrication de panneaux solaires et de la volatilité des prix des combustibles fossiles. Un inconvénient qui freine toutefois l'utilisation de l'énergie solaire est son coût d'investissement élevé. Une façon de réduire les coûts et d’augmenter la rentabilité des panneaux solaires est d'améliorer l'efficacité des panneaux photovoltaïques (PV) en termes de puissance électrique de sortie.

La tension et le courant des panneaux photovoltaïques dépendent de la température, de l'ensoleillement, de l'angle du rayonnement solaire, et d'autres conditions atmosphériques. Comme ces paramètres sont modifiés régulièrement, il est important de suivre le point de puissance maximale d'exploitation (MPOP) pour garder un maximum d'efficacité à chaque instant. Ainsi, des ajustements en temps réel de la charge externe appliquée aux panneaux photovoltaïque sont nécessaires afin de prendre en compte la puissance maximale des panneaux photovoltaïques. Dans cette recherche, la méthode

d’optimisation multi-unités est appliquée pour résoudre le problème de suivi du point de puissance maximale des panneaux photovoltaïques. Les résultats confirment la force de la méthode d'optimisation multi-unités et permettent de vérifier également le fait que les différences entre les unités peuvent être corrigées pour que chacune d’entre elles atteignent son optimum.

ABSTRACT

Optimization has become a key area in process industries due to the increasing global market competition, environmental constraints and energy costs. These factors induce operating companies to optimize plant operation in order to reduce operating cost, improve production efficiency, meet product quality specifications, and process safety. Besides, as better controllers are developed to adequately control a plant; the focus can be shifted to the solution of controller designs that guarantee optimal plant performance.

Real-time optimization (RTO) is a valuable tool, to bring and maintain a system at its optimal operating point that has received considerable attention in the process industry. Real-time optimization methods could monitor the behavior of processes, adjusting the set points of process controllers to track significant changes in the plant optimum.

Among different approaches of RTO, extremum-seeking control methods are those which are able to satisfy the necessary conditions of optimality. In other words, in extremum-seeking control methods, optimization is recast as a problem of controlling the gradient of objective function to zero. The main difference between the various extremum-seeking methods lies in the way the gradient is estimated. Most of these schemes involve injecting a periodic temporal perturbation signal and several time-scale separations are necessary to isolate the effects of the system dynamics on the estimated gradient.

Multi-unit optimization is an extremum seeking control method in which the perturbation is along the unit dimension rather than in time domain so time-scale separation is not needed and the convergence is faster for slow dynamic processes. This method requires the presence of multiple identical units, with each of them operated at input values that differ by a pre-determined constant offset. Although this method is useful when the system consist of multiple units, convergence to optimal point has been proven for systems with many identical units or two non-identical units, whereas a real industrial system model could have more than two non-identical units. Therefore, in this research, an optimization procedure based on multi-unit method is developed with respect to the number of units and number of inputs. The proposed algorithm is for the

case of a static convex objective function with two inputs. It consists of sequential corrections to compensate the differences between static surfaces of the objective functions related to each unit.

The last part of this thesis contains the case study of the multi-unit optimization method to track maximum power point of photovoltaic arrays. Electricity is mainly produced from fossil fuels, nuclear fuel and renewable resources such as sun, wind, water and biomass. Solar energy is at the forefront of clean and renewable resources and, due to advances in solar panel manufacturing and because of the volatile fuel costs, its advantage is increasing. But the actual drawback which still exists in using solar energy is its high investment cost. One way to reduce costs and increase the profitability of solar panels turns out to enhance the efficiency of photovoltaic (PV) arrays in terms of output power. The voltage and current of PV arrays depend on temperature, insolation, angle of solar irradiance, and other atmospheric conditions. As these parameters are regularly modified, it's important to track the maximum power operating point (MPOP) to keep a maximum efficiency at every instant. Thus, real-time adjustments of the external load are required to take maximum power from PV panels. In this research, multi-unit is applied as a recent technique to solve maximum power point tracking problem for PV arrays. The results confirm the strength of the multi-unit optimization method. It also verifies the fact that differences between the units can be corrected leading each of them to their respective optima.

Table of Contents

TABLE OF FIGURES

TABLE OF TABLES

LIST OF SYMBOLS AND ABREVIATIONS

RTO	Real time optimization
ESC	Extremum seeking control
NCO	Necessary conditions of optimality
NLP	Nonlinear programming
e_i	i^{th} unit vector
$\widehat{(\ )}$	Estimated variable
$\dot{(\ )}$	Derivative
$\frac{d(\ )}{d(\ )}$	Derivative
$\frac{\partial(\ )}{\partial(\ )}$	Partial derivative
Δ	Parameter of constant perturbation in multi-unit method
K	Adaptation gain
β, α	Vectors shows the differences between the optimal points of each unit
λ, ρ	Scalars shows the difference between the optimal values of each unit
$(\ )^*$	Equilibrium point
u_i^{opt}	The optimal point of unit i
J_i^{opt}	The optimal value of unit i
T_1	Period of perturbation signal for multi-unit phase
T_2	Period of perturbation signal for correction phase
PV	Photovoltaic
MPP	Maximum power point
MPPT	Maximum power point tracking
InC	Incremental conductance
P&O	Perturb and observe
CV	Constant voltage
η_{MPPT}	Efficiency of MPPT

γ	Fill factor of a PV cell
I	Output current [A]
V	Voltage across the output terminal [V]
V_{OC}	Open circuit voltage [V]
I_{SC}	Short circuit current [A]
I_{PH}	Photo current [A]
I_d	Diode current [A]
$\boldsymbol{I_S}$	PV cell's saturation current [A]
$\boldsymbol{I_{SH}}$	Shunt current [A]
I_{RS}	Diode reverse saturation current [A]
E_g	Band gap energy [eV]
K_I	PV cell's short-circuit current temperature coefficient [mA/°K]
T_{Ref}	PV cell's reference temperature [°K]
T_C	PV cell's current temperatures [°K]
G	Insolation or the intensity of solar radiation
A	Diode ideality factor (between 1 and 1.5)
q	Elementary charge (1.6021×10^{-19}C)
k	Boltzmann's constant (1.3806×10^{-23}J/°K)
P_{max}	Maximum power [W]
V_{max}	Voltage of MPP [V]
R_{L-opt}	Optimal load or the resistance related to the MPP [Ω]
I-V	Current-voltage characteristic
P-V	Power-voltage characteristic

CHAPTER 1: INTRODUCTION

1.1 Context

In the past two decades, increasing economic, quality, safety and environmental pressures have led to a greater need than ever for operating companies to explore possible paths for improving process profitability. Optimization is a key area in control theory to reduce operating cost and meet product specifications (Zhang and Forbes 2006).

Optimal operation is particularly difficult to achieve when the plant models are inaccurate or in the presence of process disturbances. In response to these difficulties, real-time optimization (RTO) has received considerable attention in the process industry. The RTO effectiveness depends on its ability to quickly and effectively identify/track the changing optimal plant operation. The ability to track changes in turn, depends on having sufficient plant information to update parameter estimates, and improving the model predictions of the process behavior (Pfaff et al. 2006).

Real-time optimization methods can be classified into two categories based on how the problem is solved: numerical or classical approach and extremum seeking control (Woodward et al. 2009a). The two-phase or the classical approach is the model-based repeated optimization where a model is adapted using the available measurements. Then, a numerical optimization is performed on the updated model. The other approach to real-time optimization is following the necessary condition of optimality along the evolution of the system; controlling the gradient to zero in unconstrained problems is such a case (Srinivasan 2007). This approach is called extremum seeking control.

Extremum seeking control is a model free optimization approach which is significantly important when there are difficulties in determining the model parameters. In this approach, optimization is recast as a problem of controlling the gradient of an objective function to zero. Differences in gradient calculation lead to different forms of extremum seeking control methods which are mainly categorized in three groups: perturbation,

adaptive extremum seeking, and multi-unit optimization. In perturbation methods, gradient is computed by applying an input perturbation and using a correlation between the input and output variations (Krstic and Wang 2000). In adaptive extremum seeking, the gradient is estimated based on a process model that is updated using available online measurements (Guay and Zhang 2003). In multi-unit optimization, the gradient is computed as a finite difference between the outputs of multiple identical units with slightly different input values (Srinivasan 2007).

In the last method, convergence to the optimal point was proven via Lyapunov analysis and it was faster than for the perturbation method. But because it was assumed that units are identical, which was a very strong assumption, in 2007, Woodward et al. analyzed a case with non-identical units (Woodward et al. 2007). They showed that for process with two non-identical units, stability is not always guaranteed and moreover the multi-unit scheme does not necessarily converge to the desired optimum. To avoid instability problem, correctors were proposed for systems with two non-identical units with one input (Woodward et al. 2009a; Woodward et al. 2010). However, real systems might have more than one input and more than two non-identical units. So, following these researchers, multi-unit optimization method is modified in this work for three non-identical units and two inputs.

To apply the developed method, photovoltaic array is chosen as a system with multiple units. The development of clean energy production has grown significantly around the world. However, several practical issues must be overcome to continue their growth. The actual drawback which still exists in using solar energy is its high cost (Cabal et al. 2007). Thus, developing methods in order to optimize the efficiency of an existing solar energy system becomes more and more important. The most readily available solar technology is the Photovoltaic (PV) array. It consists of multiple photovoltaic cells providing current-voltage (or IV) curves depending on temperature, insolation, angle of solar irradiance, and other atmospheric conditions. As these parameters are regularly changed, it’s important to track the maximum power point (MPP) to keep a maximum efficiency at every instant. When the external loads are equal to the internal resistance of

the cell, the maximum power is produced. Thus, real-time adjustments of the external load are required to produce maximum power by PV arrays.

Regarding these features, in the current research multi-unit optimization method is developed for three non-identical units and two inputs. The proposition includes a static optimization problem with a convex objective function of two variables. By means of adding correction phase to the multi-unit phase, the differences between the units in three dimensions are compensated. Besides the theoretical aspect, this method is applied for online maximum power point tracking of PV arrays.

1.2 Problem Statement

The definition of the problem under question in this thesis is as follows:

In some systems such as solar energy and wind systems, the parameter variations are fast so optimal operating point is varying. To seek for this varying optimal point there is a need for an online optimization. Real-time optimization (RTO) is a valuable tool is this area which tries to bring and maintain a system at its optimal operating point. One of the model free approaches of RTO is extremum seeking control method which has proven stability. Based on the way the gradient is estimated, extremum seeking control methods are different from each other. Multi-unit optimization is an extremum seeking control method in which the gradient is calculated based on differences between the outputs of each unit when a constant offset is introduced between the units' inputs. Although this method is useful when the system consists multiple units, convergence to optimal point has been proven provided identical units or two non-identical units.

But industrial process has more than two non-identical units and more than one input, so in this research, an optimization procedure based on multi-unit method is developed with respect to the number of units and the number of inputs. The optimization problem considered in this study is local optimization of a static and convex objective function of two variables i.e. two inputs for each unit. Besides, the multi-unit method is applied to track the maximum power point of PV arrays. This case study is chosen because of its natural configuration which consists of multi PV cells.

1.3 Main Objective

- To develop multi-unit optimization method with respect to the number of non-identical units and the number of inputs and apply it to maximize the power provided by a PV array.

1.3.1 Specific Objectives

- To develop an optimization procedure based on multi-unit method for three non-identical units and two inputs.
- To maximize output power of a PV array by applying the multi-unit method.

1.4 Structure of the Thesis

Following chapter one which includes the introduction, chapter two is dedicated to literature review and has four parts. In part one, real-time optimization (RTO) is discussed. Following that, in part two, extremum seeking control is explained as an approach of RTO. At the end of this part, a comparison between classical approach of RTO and extremum seeking control is done and multi-unit optimization is presented as a new method of extremum seeking control. Part three is about multi-unit optimization approach. Finally in part four, PV cell's modeling and output power maximization are discussed. At the end of this chapter, a brief critical analysis is done.

Third chapter presents the methodology of this research in which the objectives are defined and the overall methodology is mentioned. Then the multi-unit optimization method is explained for the case of two units and some simulation results show the functionality of this method. Following this part, development of the multi-unit algorithm for three non-identical units is discussed. The improvement in multi-unit method is obtained through the proposed adaptation laws applied to a generic mathematical example. At the end of this chapter, some guidelines for parameter tuning are addressed following by a brief conclusion of the chapter.

Fourth chapter is devoted to the application of multi-unit method to a PV array. First modeling of a PV array is described. After that the optimization problem related to PV

arrays is defined. Then the multi-unit optimization method is used to maximize the output power of PV arrays with different configurations. Both identical and non-identical cases are considered to perform multi-unit for two units and one input and simulation results are presented.

Finally, chapter five is dedicated to conclusions, and recommendations for the upcoming research works.

1.5 Contributions

Multi-unit optimization method is developed for three non-identical units and two inputs. Besides, the method is applied to the PV case study and maximum power of a PV array is achieved.

2 CHAPTER 2: RESEARCH REVIEW

2.1 Real-time optimization (RTO)

Many companies are turning to economic optimization to improve their operating efficiency and hence increase their competitive advantage in the global marketplace. Real-time optimization (RTO) is one of the tools used in this case (Darby and White 1988). The important factors which allowed optimizing process economics in *real-time* are availability of increasingly more powerful computers, improving process modeling techniques, and evolving advanced control strategies (Zhang and Forbes 2006). In any process, the optimum plant operating conditions may drift as a result of process changes. The main role of the RTO is to follow the displacement of the optimum points in the process in order to maintain the plant at its most profitable operating point. RTO effectiveness is governed by its ability to quickly and effectively identify the changing optimal plant operation at any given time (Zhang and Forbes 2000).

A schema presented by Marlin and Hrymak in 1997 showed the place of the RTO in the supervisory layer of the computer integrated manufacturing (CIM) structure and provided the bridge between plant scheduling and the control system (Marlin and Hrymak 1997).

2.2 Classical RTO structure

A typical structure of model-based RTO system approach is shown in Fig 2-1. The two-phase approach (Chen and Joseph 1987) is the most widely used method for model updating and model-based optimization in RTO.

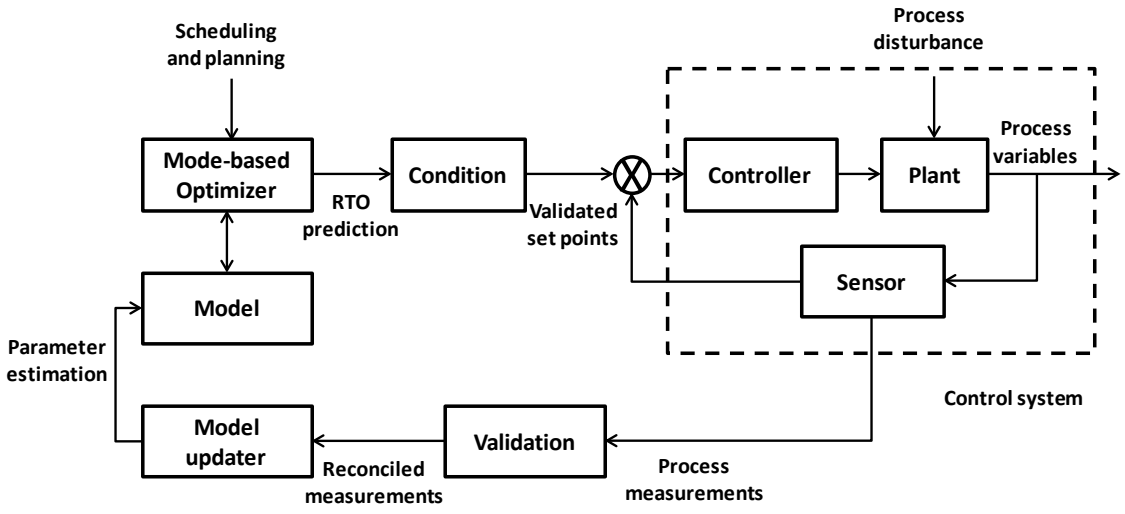

Fig 2-1: Typical structure of a model-based RTO system approach (Zhang and Forbes 2006)

The RTO loop is an extension of feedback control system and consists of subsystems for measurement validation, steady-state detection, process model updating, model-based optimization, and command conditioning (Darby and White 1988). The goal of this closed-loop adaptation is to drive the operating point towards the actual plant optimum despite of inevitable structural and parametric model mismatch (Chachuat et al. 2009). Once the plant operation reaches a steady state, plant data are collected and validated to avoid gross errors in the process measurements (White 1997).The measurements themselves might be reconciled using material and energy balances to ensure consistency of the data set used for model updating. After validation, the measurements are used to estimate the model parameters to ensure that the model correctly represents the plant at the current operating point. Then, by using the updated model, the optimum controller set points are calculated and transferred to the control system after a check by the command-conditioning subsystem (Marlin and Hrymak 1997; Sequeira et al. 2002).

2.2.1 RTO performance

The performance of RTO depends on its ability to quickly and effectively identify the changing optimal plant operation. The ability to track changes, in turn, depends on having sufficient plant information to update parameter estimates and to improve the model predictions of the process behavior (Pfaff et al. 2006). Some comprehensive

discussions of RTO technology were done in 90's decade (Marlin and Hrymak 1997; White 1997; Perkins 1998). Although the two-phase approach attempts to solve the RTO problem by updating the imperfect model, it will not necessarily converge to the correct optimum (Durbeck 1965; Biegler et al. 1985; Forbes et al. 1994). Similarly, there has been some recognition that the traditional two-step algorithm (Chen and Joseph 1987) of independent phases for model updating and model-based optimization may lead to poor RTO performance in the presence of plant/model mismatch. To address the plant/model mismatch issue, some methods have been proposed. These methods can be categorized into two classes: (1) those that modify the RTO problem directly (Roberts 1979; Becerra and Roberts 1996) and (2) those that use modified adaptive control ideas to suit RTO applications e.g. (Bamberger and Isermann 1978; Garcia and Morari 1981; McFarlane and Bacon 1989; Zhang and Roberts 1991). Although different algorithms for predicting the optimal plant operation are used in each of these methods, all of them use perturbation of the manipulated variables as a basis for compensating the plant/model mismatch. Thus, these approaches are called perturbation based methods (Zhang and Forbes 2006).

Zhang and Forbes (2000) provided a detailed discussion on RTO performance. They discussed three factors that involve in RTO system performance: (1) long term offset from the optimal plant operation, primarily caused by plant/model mismatch; (2) transmission of measurement noise; and (3) convergence characteristics (transient behavior) of the RTO system. Each of these factors depends on the process model, the model updating technique and the optimization algorithm. Besides, they proposed an RTO performance metric and design criterion called extended design cost which showed improvement in both transient and steady-state behavior of the closed-loop RTO system (Zhang and Forbes 2000). Following this work, in 2006, they did a critical performance comparison of three representative techniques from existing perturbation-based RTO methods, based on the Extended Design Cost performance criterion. Furthermore, they presented systematic methods for developing bounds on the two critical performance characteristics: convergence behavior and performance effects of required perturbations (Zhang and Forbes 2006).

In a research by Sequeira et al., 2004, the classical approach to RTO and its benefits and drawbacks were reviewed. Besides, they established a new methodology called real-time evolution (RTE) as an alternative to classical RTO or on-line model-based optimization (Sequeira et al. 2002). The difference between their proposed method and RTO lies in the fact that in RTE waiting for steady state is not necessary. Also, in 2004 they proposed a method for tuning RTE parameters (Sequeira et al. 2004).

Pfaff et al., 2006, proposed an improvement to RTO performance by integrating information generation using experimental design techniques into the RTO algorithm to reduce uncertainty in the final optimization results (Pfaff et al. 2006).

The two main causes of the RTO system not converging to the plant optimum are plant/model mismatch and uncertainty in the adjustable parameter estimates (Pfaff et al. 2006; Marchetti et al. 2009). Two main classes of optimization methods are available for handling uncertainty based on measurements availability. In the absence of measurements, a robust optimization approach is normally used whereas when measurements are available, an adaptive optimization method is preferred. Measurement-based adaptive optimization can be classified into explicit and implicit schemes, depending on whether or not a process model is used online. Fig 2-2 shows these two schemes. Explicit schemes involve two steps: first, a model update and second, numerical optimization based on the updated process model. The procedure is also called repeated optimization. These ideas have been widely discussed in the literature and used in the context of both static optimization (e.g., RTO) and dynamic optimization (e.g., model predictive control, MPC). Implicit schemes use measurements to update the inputs directly and optimality can be achieved by choosing an appropriate control structure that meets the necessary conditions of optimality (NCO). NCO tracking is formulated as a control problem that slowly moves the inputs toward the optimal solution in contrast to numerical re-optimization that provides input values that jump to the computed optimal solution. Besides, it has been shown that the use of NCO tracking (implicit scheme) can greatly simplify the implementation of optimal operation in comparison to explicit scheme using a process model (Srinivasan and Bonvin 2007).

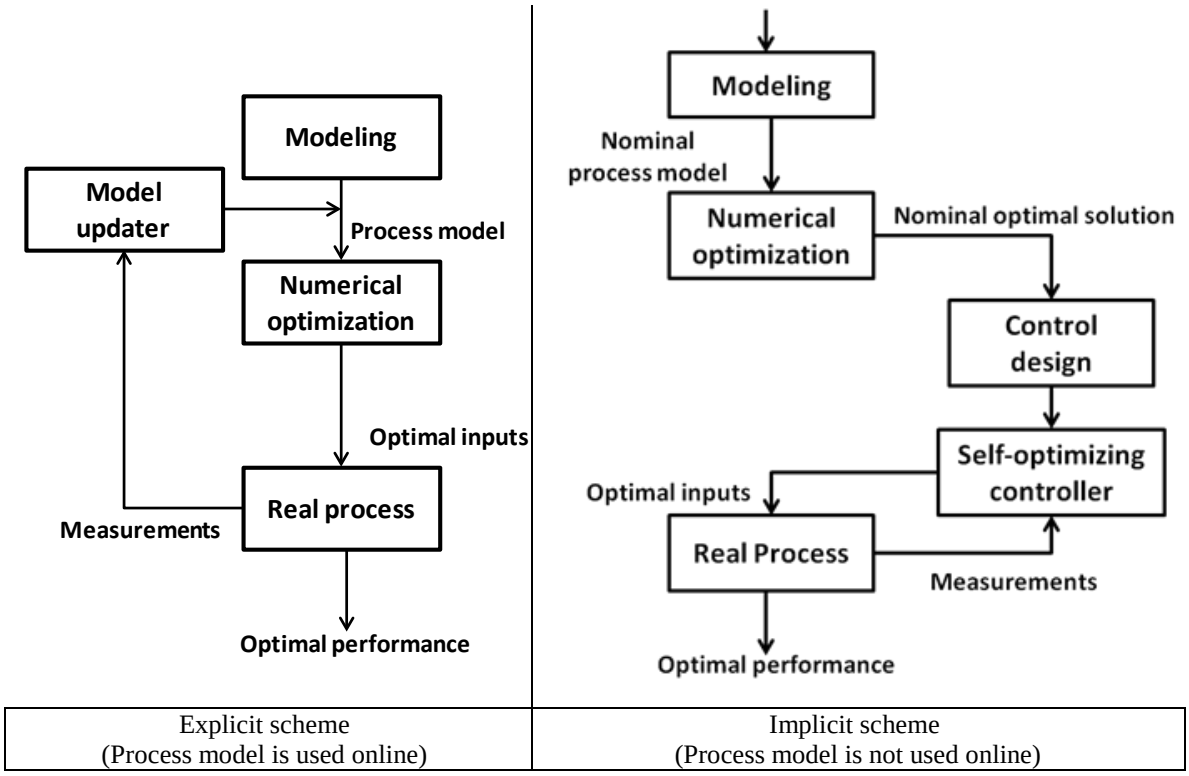

Fig 2-2: Explicit and implicit schemes (Srinivasan and Bonvin 2007)

2.2.2 Classification of real-time optimization

2.2.2.1 Classification based on type of adaptation

RTO methods can be classified in different ways. Based on the type and the objective of adaptation, it can be classified into model-parameter adaptation, modifier adaptation, and direct input adaptation (Chachuat et al. 2009). Model-parameter adaptation updates the parameters of the process model and repeats the optimization. Modifier adaptation modifies the constraints and gradients of the optimization problem and repeats the optimization. Direct input adaptation turns the optimization problem into a feedback control problem and implements optimality via tracking of appropriate controlled variables. Classification of real-time optimization approaches based on adaptation strategy, feasibility and optimality is shown in Table 2-1.The two NCO parts, namely the active constraints (related to the problem of feasibility) and the reduced gradient (related to the issue of optimality) are shown in the rows of table (Chachuat et al. 2009).

Table 2-1: Classification of real-time optimization approaches based on adaptation strategy, feasibility and optimality (Chachuat et al. 2009)

	Real-time optimization		
	Model parameter adaptation	**Modifier adaptation**	**Direct input adaptation**
Adaptation for feasibility *Constraints*	**Two step approach** (Chen and Joseph, 1987; Marlin and Hrymak, 1997)	**Bias Update** (Forbes and Marlin, 1994) **Constraint adaptation** (Chachuat et al., 2008)	**Active constraint tracking** (Maarleveld and Rijnsdrop, 1970; Srinivasan et al.,2001)
Adaptation for optimality *Constraints and gradients*	**Two step approach** (Chen and Joseph, 1987; Marlin and Hrymak, 1997) **Identification for optimization** (Srinivasan and Bonvin,2002)	**Integrated system optimization and parameter estimation (ISOPE)** (Roberts, 1970; Tatjewski, 2002; Brdys and Tatjewski, 2005) **Gradient correction** (Gao and Engell, 2005; Marchetti et al., 2009)	**Self-optimizing control** (Skogestad, 2000; Govatsmark and Skogestad, 2005) **Extremum seeking** (Ariyur and Krstic, 2003) **NCO tracking** (Francois et al., 2005; Srinivasan et al., 2008)

2.2.2.2 Classification based on the presence of a model

Based on presence of model, RTO methods can be classified into model-based approach, fix model approach, and model free approach. The classical approach is a model-based approach that consists of model adaptation using available measurements and numerical optimization which is performed on the updated model. So, a wealth of literature has been devoted to model-based RTO e.g. (Marlin and Hrymak 1997; Zhang et al. 2002). The classical two-step approach works well when there is little structural plant/model mismatch, and the changing operating conditions provide sufficient excitation for estimating the uncertain model parameters. As these conditions are rarely met in practice, fixed-model and model-free methods which do not rely on model-parameter update have gained in popularity recently (Marchetti et al. 2009).

Fixed-model methods use both a nominal process model and appropriate measurements to find optimal point by an iterative scheme. The process model is embedded within a nonlinear programming (NLP) problem that is solved repeatedly. However, the measurements are used to update the cost and constraint functions in the optimization problem instead of refining the parameters of a first principles model from one RTO's

iteration to the next. Thus, it achieves a better approximation of the plant cost and constraints at the current point (Forbes and Marlin 1994; Gao and Engell 2005; Chachuat et al. 2008; Chachuat et al. 2009).

Model-free methods do not use a process model online to implement the optimization. These methods can be classified into two approaches. In the first one, successive operating points are determined by mimicking iterative numerical optimization algorithms e.g. (Box and Draper 1969; Garcia and Morari 1981). The second approach to model-free methods consists in recasting the NLP problem into that of choosing outputs whose optimal values are approximately invariant to uncertainty e.g. (Skogestad 2000; François et al. 2005). The second approach involves directly meeting the NCO along the evolution of the system and it treats the optimization problem as a control problem with all the advantages related to sensitivity reduction and disturbance rejection (Srinivasan 2007; Marchetti et al. 2009). This model-free optimization method has been studied under the name of extremum seeking control, where the basic concept is to reformulate the unconstrained optimization problem as a problem of controlling the gradient of the objective function to zero. The method is quite old (Leblanc 1922) but it has received renewed interest recently (Ariyur and Krstic 2003; Guay and Zhang 2003; Srinivasan 2007). Also, many recent publications have reported successful applications (Ariyur and Krstic 2003; Popovic et al. 2006).

2.3 Extremum seeking control

The main methods of adaptive control (both linear and nonlinear) deal only with regulation to *known* set points or reference trajectories (Landau 1979; Krstic et al. 1995; Ioannou and Sun 1996; Khalil 2002). But the control objective could be to optimize an objective function which can be a function of *unknown* parameters, or to select the desired states to keep a performance function at its extremum value. Self optimizing control and extremum seeking control (ESC) are two methods to handle these kinds of optimization problems.

Finding a set of controller variables, when kept at constant set-points, which indirectly lead to near-optimal operation with acceptable loss, is the goal of self-optimizing control

(Findeisen et al. 1980; Morari et al. 1980; Skogestad 2000). Finding the operating set-points that maximize or minimize an objective function is the task of extremum seeking (Guay and Zhang 2003). Based on Astrom definition, ESC is tracking a varying maximum or minimum of an output (performance) function (Astrom and Wittenmark 1994) which has two layers of meaning: first, seeking an extremum of the output function; secondly, ability to control (stabilize) the system and drive the output to that extremum.

The early research work on extremum seeking control was in the 1920's (Leblanc 1922). Extremum seeking control and self-optimizing control were popular in the 1950s and 1960s, much before the theoretical breakthroughs in adaptive linear control of the 1980s. Besides, many successful applications of extremum seeking control approaches have been reported, for example, combustion process control for IC engines and gas furnaces (Sternby 1980; Astrom and Wittenmark 1994), and anti-lock braking system control (Drakunov et al. 1995).

The uncertainty associated with the objective function in ESC makes it necessary to use some sort of adaptation and perturbation to search for the optimal operating conditions. Thus, most of ESC schemes involve injecting a periodic temporal perturbation signal. A systematic description of the perturbation based extremum seeking control and its applications were presented in Ariyur and Krstic 2003. Extremum seeking control via perturbation method by Krstic and Wang 2000 considered a general SISO nonlinear model $\dot{x} = f(x,u)\ and\ y = h(x)$ where $x \in R^n$ is the state, $u \in R$ is the input, $y \in R$ is the output, and $f: R^n \times R \to R^n$and $h: R^n \to R$ are smooth. They supposed a known smooth control law $u = \propto (x,\theta)$ parameterized by a scalar parameter θ. Extremum seeking control via perturbation method is shown in Fig 2-3 (Krstic and Wang 2000).

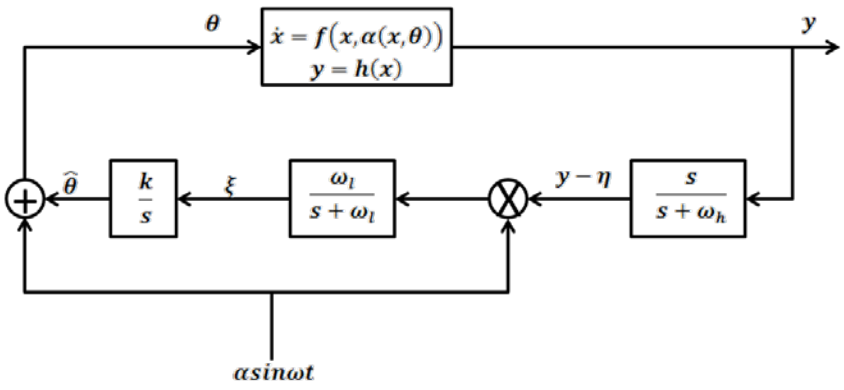

Fig 2-3: Extremum-seeking control via perturbation method (Krstic and Wang 2000)

In the method using perturbations, a temporal perturbation (or dither) is injected along the input and the gradient is estimated by using the correlation between the input and the output. An integral controller is used to control the gradient to zero. The multiplication with the perturbation is performed to capture the output that is correlated with the input and a low pass filter is used to take the average of the oscillations. A high pass filter at the output of the system is used to remove the bias. Several time scale separations are necessary to isolate the effects of the system dynamics on the estimated gradient. The three time scales consist of fastest (for the plant with the stabilizing controller), medium (for the periodic perturbation), and slow (for the filters in the peak seeking scheme) (Krstic and Wang 2000).

The first rigorous proof of local stability of perturbation based extremum seeking control scheme was presented by Krstic and Wang 2000. They used averaging analysis and singular perturbation, where a high pass filter and slow perturbation signal were employed to derive the gradient information. Their proof covered only one form of extremum control (the method with a periodic perturbation) (Krstic and Wang 2000), besides, the plant had to be very fast (quasi-static) and the adaptation gain had to be small which means the conditions imposed were restrictive. Following this work, Krstic presented a tighter analysis which removed these conditions. He proposed dynamic compensation to provide stability guarantee, fast tracking of changes in the operating point, and measurement noise rejection (Krstic 2000). This method is limited to the problems with linear dynamics and it is not useful in other cases.

In 2002, Choi proposed an extremum seeking control algorithm for discrete-time systems applied to a class of plants that are represented as a series combination of a linear input dynamics, a static nonlinearity with an extremum, and a linear output dynamics. They used the two-time scale averaging theory to derive a mild sufficient condition under which the plant output exponentially converges to a neighborhood of the extremum value (Choi et al. 2002).

Recently, several extremum control schemes and stability analysis for extremum seeking of linear unknown systems and a class of general nonlinear systems are presented (Krstic 2000; Krstic and Wang 2000; Wang et al. 2000). This framework allowed the use of black-box objective functions with the restriction that the objective value is an available output for online measurement. Although this technique was proven useful for some applications (Krstic et al. 1999; Nguang and Chen 2000; Wang et al. 2000), the lack of guaranteed transient performance of the black-box schemes remained a significant drawback in its application.

Alternatively, in 2003, Guay and Zhang used an adapted model of the system for analytical evaluation of the gradient (Guay and Zhang 2003). Their extremum seeking framework assumes that the objective function is explicitly known as a function of the system states and uncertain parameters from the system dynamic equations. Only an estimated value based on parameter estimates is available because parametric uncertainties make the on-line reconstruction of the true cost impossible. The control objective was to simultaneously identify and regulate the system to the extremum point, which depends on the uncertain parameters (Guay and Zhang 2003). The main advantage of their proposed approach was that some degree of transient performance can be guaranteed, and the optimization objectives were achieved when a reasonable functional approximation of the objective function was available. In 2004, Dehaan and Guay extended this approach to nonlinear systems with unknown parameters whose states must satisfy a set of known convex constraints (DeHaan and Guay 2004). Then in 2005, they generalized the approach of Guay and Zhang (2003), and DeHaan and Guay (2004) to include systems whose states must satisfy a set of known convex inequality

constraints and they achieved a nominal guarantee of transient performance by using a Lyapunov-based approach (DeHaan and Guay 2005).

Adelota et al., in 2004 presented a control algorithm that incorporated real time optimization and receding horizon control technique in order to solve an extremum seeking control problem for a class of nonlinear systems with parametric uncertainties (Adetola et al. 2004). In 2006, Adelota and Guay proposed a control algorithm which was an integration of real-time optimization and model predictive control to solve an output feedback extremum seeking control problem for a linear unknown system. The resulting controller could drive the system states to the desired unknown optimum by requiring a Lyapunov restriction and a satisfaction of a persistency of excitation condition (Adetola and Guay 2006). They discussed the problem of parameter convergence in adaptive extremum seeking control design in 2007. They proposed an alternate version of the popular persistence of excitation condition for a class of nonlinear systems with parametric uncertainties. Parameter convergence with minimal but sufficient level of perturbation was guaranteed by their presented method (Adetola and Guay 2007).

Banavar in 2003 solved the extremum seeking control problem by assuming that the performance function can be approximated by a quadratic function with a finite number of parameters which were estimated on-line. In contrast to traditional approaches, time-scale separation between the gradient computation and function minimization and the system dynamics was not needed. A significant advantage of a quadratic function is that it allows the peak-seeking control loop to be reduced to a linear system. For such a loop, the wealth of linear system analysis and synthesis tools can be employed (Banavar 2003).

Zhang and Ordonez in 2005 proposed an extremum seeking control scheme for linear time invariant (LTI) systems. The convergence and robustness of the extremum seeking scheme were guaranteed by the numerical optimization algorithm, and also a detailed analysis based on the line search method was addressed (Zhang and Ordónez 2005). Following the research on numerical optimization-based extremum seeking control, they

proposed an extremum seeking via a state regulator that drove the state traveling along a convergent set point sequence generated by a numerical optimization algorithm (Zhang and Ordónez 2006).

Chioua et al., in 2007 showed that in some particular situations the perturbation based extremum seeking algorithm may not converge to the optimum but only close to it. The error for a general nonlinear dynamic system is proportional not only to the square of the excitation amplitude but also to the square of the frequency of excitation. They addressed that slower optimization frequency is not only required for stability purposes but also for accuracy. As a conclusion, they showed that the frequency of excitation should be low which in turn makes the optimization slower if accuracy is required (Chioua et al. 2007).

Most of extremum seeking schemes uses deterministic periodic perturbations, but periodicity can naturally lead to predictability which is not desirable in cases like some tracking and navigation applications. As a solution to these problems, in 2009, Manzie and Krstic proposed a method of extremum seeking by using stochastic perturbation. Convergence towards the extremum of a static map can be guaranteed with their stochastic extremum seeking algorithm. Besides, they quantified the behavior of a system with Gaussian-distributed perturbations at the extremum in terms of the extremum seeking constants and map parameters (Manzie and Krstic 2009).

Based on the literature, in Table 2-2 a brief comparison between classical approach and extremum seeking control as two main classes of RTO is presented. Both advantages and disadvantages are shown. Extremum seeking control is a model free approach which involves directly meeting the NCO. NCO-tracking scheme helps link the framework of *measurement-based optimization* to the fields of *identification and control* (Srinivasan 2007). So, the numerous tools available in the context of identification and control can provide the mathematical framework necessary for the analysis and design of extremum seeking control.

Table 2-2: Comparison between classical RTO and extremum seeking control

Method	Numerical or classical approach	Extremum seeking control
Advantages	• Rapid Convergence • Apply to large problems (Woodward 2009) • Allow to handle the constraints more directly (Woodward et al. 2007)	• Model free • Proper accuracy (Woodward 2009) • Feedback loop filters the measurement noise • More robust to noise by tuning the integral gain • Sensitivity reduction and disturbance rejection
Disadvantages	• Poor precision • Plant model mismatch • Identification of model parameters affected by noise measurements • Computationally intensive (Woodward 2009)	• Slow convergence • Impractical in large problems • Wait till the dynamics die down before the gradient can be computed • Experimentally expensive (Woodward 2009)

2.3.1 Classification of ESC methods

Classification of extremum seeking control methods is based on the method of gradient estimation. Several techniques for estimating the plant gradients have been proposed, which differ in terms of their relying on a model or not, as well as their use of steady-state vs. transient measurement data. Three main classes of ESC methods are perturbation based, adaptive extremum seeking, and multi-unit optimization (Woodward et al. 2007).

Perturbation methods (Leblanc 1922; Krstic and Wang 2000), requires direct measurement of the cost function. They use an input perturbation and compute the gradient using a correlation between the input and output variations. In adaptive extremum seeking method (Guay and Zhang 2003), additional measurements and not necessarily the cost function are needed . Gradient is calculated based on a process model that is updated using available on-line measurements. In multi-unit optimization (Srinivasan 2007) the gradient is computed as a finite difference between the outputs of multiple units with slightly different input values.

The main difficulty of perturbation method (Krstic and Wang 2000) is the requirement of a multiple time-scale separation between the system dynamics, the perturbation frequency and the adaptation rate. The perturbation has to be an order of magnitude slower than the system dynamics to separate the effect of the perturbation from that of system dynamics. Also, the adaptation dynamics should be another order of magnitude slower in order to distinguish the effect of the perturbation from that of adaptation. This multiple time-scale separation leads to slow convergence. Time-scale separation is not an issue for processes with fast responses, e.g. electrical or mechanical systems, though, for slower processes such as the chemical or biological ones, the convergence time could be prohibitive. Another problem with perturbation method is that the output is not in phase with the input due to the system dynamics. This phase shift will cause the scheme to converge elsewhere from the optimum (Srinivasan 2007)

Adaptive extremum seeking (Guay and Zhang 2003) and multi-unit optimization (Srinivasan 2007) methods were proposed in response to limitation of perturbation based methods.

2.4 Multi-unit optimization

Multi-unit method was proposed by Srinivasan in 2007. This scheme required the presence of multiple identical units, with each of them operated at input values that differ by a pre-determined constant offset. Micro array reactors, production lines and photovoltaic arrays are examples of such system (Srinivasan 2007).

In Fig 2-4 the schematic of multi-unit optimization is shown. The system has m+1 identical units, where m is the dimension of input of the system. The optimization problem is formulated considering a dynamic system with state $x \in R^n$, and input $u \in R^m$. This system has to be operated to minimize a convex function:

$$\min_u J(u,x)\, s.t. \dot{x} = F(x,u) \equiv 0 \qquad \text{Eq 1}$$

where $F(x,u)$ is the function describing the dynamics of the system, which is assumed to be stable. The necessary conditions of optimality are given by:

$$\frac{dJ}{du} = \frac{\partial J}{\partial u} - \frac{\partial J}{\partial x}\left(\frac{\partial F}{\partial x}\right)^{-1}\frac{\partial F}{\partial u} \quad \text{Eq 2}$$

The various units are operated with input values that are slightly different. The first unit is the reference and is operated at the input value u_0 . For the other unit

$$\boldsymbol{u_i = u_0 + e_i\Delta} \quad \text{Eq 3}$$

where e_i is i^{th} unit vector and $i = \{1,2,\dots,m\}$.

The gradient is estimated by

$$\widehat{\boldsymbol{g_i}}(\boldsymbol{u_0}) = \frac{\bar{J}(x_i - u_i) - \bar{J}(x_0 - u_0)}{\Delta} \quad \text{Eq 4}$$

$$\left(\frac{\widehat{dJ}}{du}\right) = \widehat{\boldsymbol{g}} \quad \text{Eq 5}$$

and $\hat{g}_i$ is the i[th] row of gradient vector $\hat{g}$. The extremum seeking control law is:

$$\dot{\boldsymbol{u}}_{\boldsymbol{i}} = -\boldsymbol{k}\widehat{\boldsymbol{g}}^{\boldsymbol{T}}(\boldsymbol{u_0}) \quad \text{Eq 6}$$

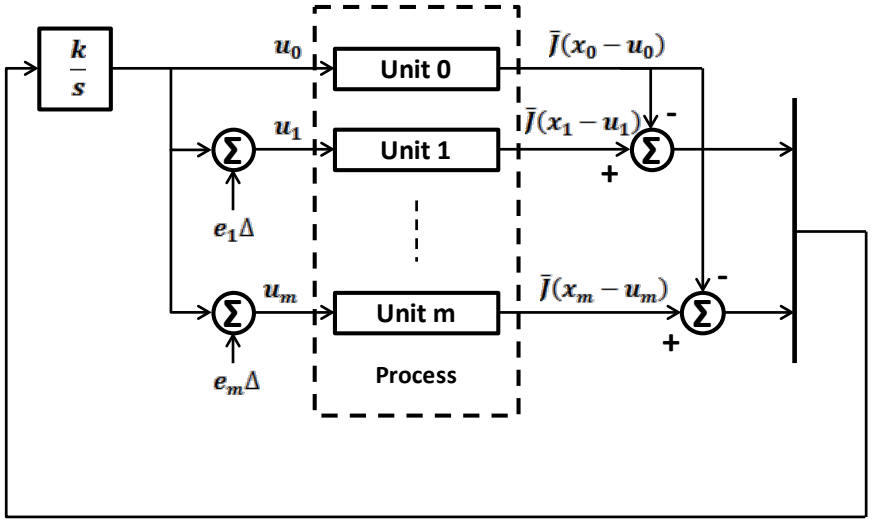

Fig 2-4: Schematic for multi-unit optimization (Woodward et al. 2007)

In his work, the gradient was computed as a finite difference between the outputs or the objective functions related to multiple units with slightly different input values (Srinivasan 2007). Thus, the perturbation was along the unit dimension rather than in time domain. So, time-scale separation was not needed and the convergence was faster. Also, he established the convergence of this method rigorously under certain assumptions (e.g. the convexity) by Lyapunov analysis.

The multi-unit optimization presented by Srinivasan in 2007 required the presence of multiple identical units which was a very strong assumption and might not be realizable in practice. So in 2009, Woodward et al., studied the effects of the differences between the static characteristics on the stability and convergence of the standard multi-unit optimization scheme. For processes with non-identical units, it was shown that differences in the static characteristics could lead the equilibrium point to be quite far away from the desired optimum. Furthermore, they proved that convergence conditions can be satisfied in two different ways: by choosing the correct sign or a large enough value for Δ. While the second option is hard to quantify, the sign adjustment could be made possible if auxiliary information is available (Woodward et al. 2009a).

To avoid the situation in which the equilibrium point is far away from the desired optimum, they proposed correctors which compensate for the differences between the units. Two types of adaptation were analyzed: a sequential approach (Woodward et al. 2009a) where the multi-unit adaptation and the correction were done separately and a simultaneous approach (Woodward et al. 2010) where both were performed together. In both cases they showed that the scheme with correctors is locally asymptotically stable and converges to the respective optimum of each unit. In both approaches, they considered the single input and two similar units' case to simplify the presentation of the method. The units had the same curvature but were shift in "u" and "J" dimensions, so on one hand they are identical since they have the same static curve, on the other hand, they are different since their optimal point of operation are not the same. Besides, they assumed that the dynamics are the same and are very fast compared to the optimization time-scale so the process can be considered quasi-static. Also no noise effects were considered, and the functions were convex.

In *sequential approach*, correctors $\hat{\beta}$ and $\hat{\gamma}$ attenuate the effect of the differences in the optimal points of operation, and in the optimal values of the performance function respectively. By alternation between the multi-unit method and the calculation of correctors with two different perturbation signals they derived the update laws for the estimates (Woodward et al. 2009a; Woodward et al. 2009b). In sequential approach, optimization and correction for differences are performed alternatively which causes a discontinuous operation leading to a hybrid dynamics. To avoid such a scenario, an approach where optimization and correction take place simultaneously, is presented by Woodward et al., 2010. In the simultaneous approach, the correctors $\hat{\beta}$ and $\hat{\gamma}$, are adapted simultaneously with the evolution of the process to its optimum. Structure of this method is shown in Fig 2-5.

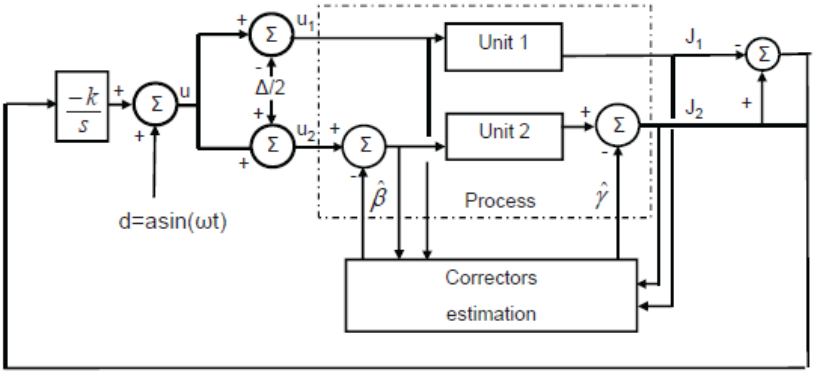

Fig 2-5: Structure of the multi-unit optimization method with simultaneous adaptive correctors (Woodward et al. 2010)

Summary of this method is shown in Table 2-3.

Table 2-3: Summary of simultaneous correction for two non-identical units

Synchronization of the two units:	$u_1 = u - \frac{\Delta}{2} + a\sin\omega t$ $u_2 = u + \frac{\Delta}{2} + a\sin\omega t - \hat{\beta}$
Multi-unit adaptation law:	$\dot{u} = -\frac{k_{mu}}{\Delta}(J_2 - J_1 - \hat{\gamma})$
Adaptation laws for correctors:	$\dot{\hat{\beta}} = -k_\beta(\hat{u}_1^{opt} - \hat{u}_2^{opt} - \hat{\beta})$ $\dot{\hat{\gamma}} = -k_\gamma(J_2(\hat{u}_2^{opt}) - J_1(\hat{u}_1^{opt}) - \hat{\gamma})$

Besides proposing the correctors, Woodward did experimental verification of the multi-unit optimization method for the maximum power point tracking of microbial fuel cells. The sequential adaptation technique was used to correct the difference between the cells. The experimental results confirmed the main advantage of the multi-unit optimization method, i.e., a faster convergence to the optimum than methods using temporal perturbation. Moreover it verified the fact that differences between the units can be corrected leading each of them to their respective optima (Woodward et al. 2009a; Woodward et al. 2009b).

Although lots of successful attempts have been made to study multi-unit optimization for non-identical units, all of them assumed only two non-identical units but industrial process has more than two units and more than one input.

2.5 Photovoltaic cells

Solar energy is at the forefront of clean and renewable resources and according to advances in solar panel manufacturing and increasingly volatile fuel costs, its advantage is rising. The major advantages of using PV system are short lead time for designing and installing a new system, output power matching with peak load demands, static structure, no moving parts, longer life, no noise, high power capability per unit of weight, inexhaustible and pollution free, highly mobile and portable because of its light weight (Krauter 2006; Petreuş et al. 2008; Tsai et al. 2008). But the actual drawback which still exists in using solar energy is its high cost (Cabal et al. 2007). One way to diminish cost and increase the profitability of solar panels is efficiency enhancement in terms of output power. Photovoltaic (PV) cell is the basic device that generates electricity when exposed to light. The structural parts of solar energy system -from smallest to largest- are PV cells, PV modules or PV panels, and PV array.

2.6 PV arrays

The single-diode model is the best model fitted for the mono-crystalline PV cell which has best efficiency among all commercially available technology. But for other competitive technology of same class (e.g. polycrystalline), two-diode equivalent circuit

or double exponential model is fitted more properly (Nema et al. 2009). In the other hand double exponential model is rarely used in the subsequent literatures because of some limitations to develop expressions for the I-V curve parameters subject to the implicit and nonlinear nature of the model (Tsai et al. 2008).

Petreuş et al., 2008, presented four models for a photovoltaic cell. They evaluated each model and identified their strengths/weaknesses. The one-diode model, the two-diode model, the first empirical model, and the second empirical model were investigated (Petreuş et al. 2008).

The mathematical models are more fitted to physics of photovoltaic cells than empirical models because they are based on the theoretical equations that describe the operation of the photovoltaic cells. A general mathematical description of I-V output characteristics for a PV cell has been studied for over the past four decades (Tsai et al. 2008). Several researchers used single-diode model for their studies on PV cell (Hussein et al. 1995; Joyce et al. 2001; Cabal et al. 2007; Tsai et al. 2008; Nema et al. 2009; Villalva and Gazoli 2009; Nema et al. 2010; Chiu et al. 2011) and etc. The schematic of a single-diode model is shown in Fig 2-6. This equivalent circuit consists of a photo current, a diode, a parallel resistor expressing a leakage current, and a series resistor describing an internal resistance to the current flow.

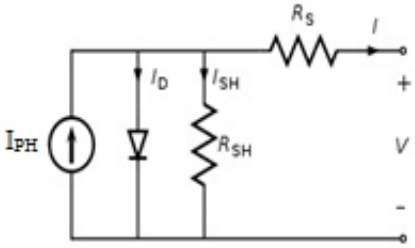

Fig 2-6: Single-diode model

Since a typical PV cell produces less than 2W at 0.5V-0.8V (depending on the cell technology) approximately, the cells must be connected in series-parallel configuration to produce enough voltage and power (Tsai et al. 2008; Nema et al. 2009).

Solar photovoltaic system consists of PV array, inverter, energy storage (e.g. batteries), system charge control, load, and balance of system components. There are two major types of PV systems: stand alone (off Grid), and grid connected showed in Fig 2-7.

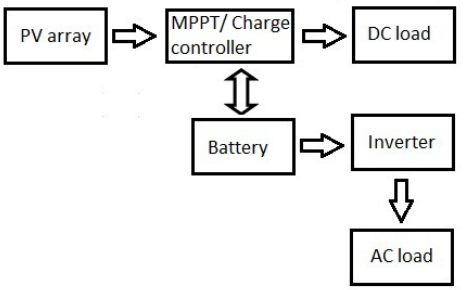

Diagram of stand-alone PV system with battery storage powering DC and AC loads

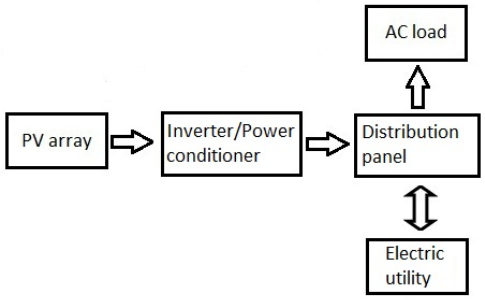

Diagram of grid-connected photovoltaic system

Fig 2-7: Schematic of two general types of PV system

The main function of the solar regulator or solar controller is to keep batteries fully charged. The solar controller regulates the flow of electricity from a solar panel to the battery without allowing the battery to be overcharged and at the same time preventing current flowing back from the battery to the solar panel. PV arrays should be used in conjunction with Deep cycle batteries. These batteries are designed to be charged and discharged over a long period of time. They are not the same as car batteries which provide a large amount of current for a short period of time. The lead-acid battery has low cost and high capacity features and is widely used in various applications such as uninterruptible power system (UPS), automotive power system and telecom power supply, but they have some disadvantages such as poor energy density characteristics, charging time and lifetime (Bright-Green-Energy 2009; Chiu et al. 2011).

Few works have been done in the literature, e.g. (Joyce et al. 2001), for modeling a PV system consisting of PV array, charger, and batteries, though Joyce et al., 2001, proposed models for PV array, batteries, charger, and inverter.

2.7 Maximum power point tracking

In Fig 2-8, the I-V and P-V characteristic of a PV array is shown. The most real attainable power is defined by the greatest possible of voltage and current at an operating point, which is called maximum power point (MPP).

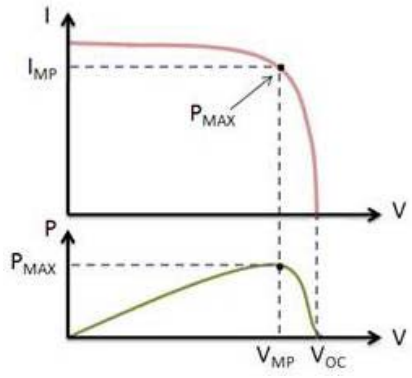

Fig 2-8: I-V and P-V characteristics of a PV cell

The maximum power can be expressed as:

$$\boldsymbol{P_{max} = V_{max} I_{max} = \gamma V_{OC} I_{OC}}$$
Eq 7

where V_{max} and I_{max} are terminal voltage and output current of PV module at maximum power point (MPP), and γ is the cell fill factor which is a measure of cell quality.

Based on Jacobi's law, a power source will deliver its maximum power to a load when the load has the same impedance as the internal impedance of the power source. Unfortunately, batteries are far from the ideal load for a solar array and the mismatch results in major efficiency losses. Maximum power point tracking (MPPT) is designed to overcome this problem. MPPT presents an ideal load to the PV array allowing it to operate at its optimum voltage. A variable DC/DC converter in the module automatically adjusts the DC output from the module to match the battery voltage (Electropaedia 2005; Petreuş et al. 2008). The output current and power of PV cell depend on the cell's terminal operating voltage along with temperature, insolation, angle of solar irradiance, shading, and other atmospheric conditions. For example, with increase of working temperature, the short-circuit current of the PV cell increases, whereas the maximum

power output decreases. On the other hand, with increase of insolation, the short-circuit current of the PV module increases, and the maximum power output increases as well (Tsai et al. 2008).

However it is not enough to match the voltage at the specified maximum power point (MPP) of the PV array to the varying battery voltage as the battery charges up. Due to changes in atmospheric condition, the MPP of the PV also changes. Thus, there is a moving reference point and a moving target. For optimum power transfer, the MPPT system needs to track the MPP as the temperature and insolation changes in order to provide a dynamic reference point to the voltage regulator (Electropaedia 2005)

In general, the maximization of the power supplied by PV panel is carried out by two main methods: mechanical and/or electronic systems (Leyva et al. 2006). Mechanical methods are based on the improvement of the irradiance conditions on solar cells (e.g. sun tracking and reduction of optical reflections) and/or on the temperature reduction during cell operation (e.g. use of cooling device). Electronic/electrical methods are based on changing load to the optimum load which leads to track maximum power at each moment, e.g. perturb and observe (P&O) algorithms (Wasynezuk 2007), Incremental conductance (InC) (Hussein et al. 1995), constant voltage and current (CV) (Andersen and Alvsten 1995), pilot cell algorithm (Salameh et al. 1991), parasitic capacitance (Brambilla et al. 1999), model-based algorithms (Bohórquez et al. 2009) , fuzzy methods (Won et al. 1994), algorithms based on digital signal processing (Hua et al. 2002), RTO based on extremum seeking methods (Leyva et al. 2006), adaptive digital MPPT based on extremum seeking control (Cabal et al. 2007).

According to literature, some of the most prominent MPPT methods for PV systems are presented in this part. Most of these methods used Eq 8 for evaluating the MPPT efficiency, where P_{actual} is the actual or measured power produced by the PV array under the control of the MPPT, and P_{max} is the true maximum power the array could produce under a given temperature and irradiance.

$$\boldsymbol{\eta_{MPPT}} = \frac{\int_0^t \boldsymbol{P_{actual}(t)}}{\int_0^t \boldsymbol{P_{max}(t)}} \qquad \text{Eq 8}$$

Hussein et al., in 1995, developed an MPPT algorithm, named incremental conductance or InC. Both results from simulation and experiment showed successfully tracking the MPP even in cases of rapidly changing atmospheric conditions and had higher efficiency than ordinary algorithm, such as perturb and observe (Hussein et al. 1995).

In 2002, Hohm et al., compared between the efficiencies of some MPPT algorithms. Their experimental results showed 97.8% efficiency for P&O (after properly optimized), 97.4% efficiency for InC, and 91.2% for constant voltage (CV) methods. They found that the P&O method could be highly competitive against other MPPT algorithms. Incremental conductance performed as well as P&O, but in general it has higher implementation cost (Hohm and Ropp 2003).

An MPPT system based on extremum seeking control was developed by Leyva et al., in 2006. The MPPT guaranteed the stability of the maximum seeking procedure for large-signal operation and the theoretical predictions were experimentally validated in a PV system (Leyva et al. 2006).

An adaptive digital MPPT based on extremum seeking control was developed by Cabal et al. in 2007. They implemented the extremum seeking control in the PIC18F1220 microcontroller. They achieved a high quality matching between sources and loads by adjusting continually the static converter duty cycle. The control of the converter through its duty cycle allowed tracking the MPP when the PV was exposed to the climatic variation. This system had a high efficiency in steady state but also during transitory. Results showed the solar panel efficiency of almost 99% (Cabal et al. 2007).

Some works focused on the control of grid-connected photovoltaic arrays e.g. (Bratcu et al. 2008; Azevedo et al. 2009). The global scope of tracking the maximum power point under variable conditions of irradiance was achieved by using a simple and robust P&O extremum seeking control scheme (Bratcu et al. 2008). In 2009, Azevado et al., showed that the P&O and InC techniques could be improved through the optimum adjustment of the sampling rate and perturbation size both in accordance with the converter dynamics (Azevedo et al. 2009).

A battery charger with MPPT function for low-power PV system applications was presented in a study by Chiu et al., 2010. The operation and design considerations of the proposed PV charger were discussed in detail. Experimental results showed that high MPPT accuracy and conversion efficiency can be simultaneously achieved under high-frequency operation (Chiu et al. 2011).

In 2010, Enrique et al. developed a method as an analog version of the P&O-oriented algorithm. They stated that this method maintains P&O main advantages such as simplicity, reliability, low price and easy practical implementation, and avoids P&O main disadvantages like inaccuracy and relatively slow response. Once the system has reached the MPP, the efficiency is superior to 99%, improving the ones obtained by other methods (P&O, InC, CV) (Enrique et al. 2010).

2.8 Summary

In the first part of this chapter, RTO, its performance, and classification were presented. RTO is a valuable tool to bring and maintain a system at its optimal operating point that has received considerable attention in the industry. General properties of two main classes of RTO methods have been showed in Table 2-2 and pros and cons of each one have been analyzed. As it has been shown extremum seeking control approach has some advantages over classical approach of RTO such as proven stability in convergence to optimal point and its model free properties.

In the second part, extremum seeking control method was explained as a powerful approach of RTO. Besides, a classification of ESC has been presented. Among extremum seeking methods, multi-unit optimization has some properties related to the way in which gradient is calculated such as faster convergence to optimal point.

Third part has been dedicated to multi-unit method. Although this method is useful when the system consist multiple units, convergence to optimal point has been proven provided identical units or two non-identical units, whereas a real industrial system model could have more than two non-identical units. So in this research, an optimization procedure based on multi-unit method will be developed with respect to the number of units and the

number of inputs. The optimization problem considered in this study is the local optimization of a static and continuous system, where the objective function is convex.

In the last part, PV cell and array models were introduced. Besides, the maximum power point tracking problem was presented to show the importance of solving this optimization problem in PV arrays. Because of the configuration of PV cells in a PV array, this system is a proper test bed for applying the multi-unit algorithm. Besides, the values of PV cell parameters are not known certainly and can vary between cells from the same production run (Hohm and Ropp 2003), so model-based MPPT are not practical and multi-unit method, as a model free one, has some benefits over model-based methods. In most of algorithms of extremum seeking method MPPT, the stability has not been analytically proved (Leyva et al. 2006) but in multi-unit method using correctors, in contrast, the stability for two non-identical units has been analytically proven via Lyapunov approach (Woodward et al. 2009a).

3 CHAPTER 3: METHODOLOGY -LOCAL OPTIMIZATION WITH MULTI-UNIT METHOD FOR QUADRATIC OBJECTIVE FUNCTIONS

3.1 Main Objective

- To develop multi-unit optimization method with respect to the number of non-identical units and the number of inputs and apply it to maximize the power provided by a PV array.

3.2 Specific Objectives

- To develop an optimization procedure based on multi-unit method for three non-identical units and two inputs.
- To maximize output power of a PV array by applying the multi-unit method.

3.3 Overall Methodology

To achieve the main objective, two specific objectives are defined. For the first specific objective or to develop the multi-unit optimization procedure for three units and two inputs, the overall methodology includes three steps which are shown in Fig 3-1. Step one is explaining the multi-unit method for two units and one input to display how the algorithm works generally for identical units. Then the idea of correction is investigated to show its ability to make multi-unit converges to the relative optimal points in the case of two non-identical unit. In step two, first the multi unit is applied for three identical units and two inputs and the optimization problem is introduced for an objective function of two variables. After that the problem of using multi-unit algorithm for three non-identical units and two inputs is presented. Moreover the extension of correction phase for three units is expressed and the adaptation laws and the schematic of multi-unit optimization in the case of three non-identical units are proposed. Following this part, the functionality of the developed algorithm is verified by applying it on a generic mathematical function. The third step consists of some guidelines to tune the parameters and gains in the whole procedure of multi-unit algorithm which is profitable for any user of multi-unit optimization algorithm. In other words, the priorities in tuning the parameters are discussed. At the end of this chapter a brief conclusion is mentioned.

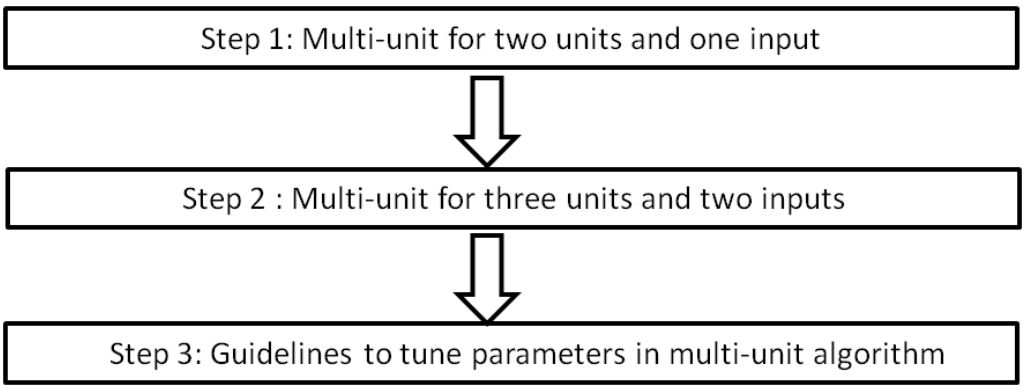

Fig 3-1: Overall methodology

3.4 Multi-unit optimization for two units and one input

As it is presented in chapter one, in multi-unit optimization gradient is calculated base on the differences of the outputs of the units which have slightly different inputs. In this part both cases with two identical/non-identical units are explained via illustrative examples and the simulation results display how this method works.

3.4.1 Identical units

Fig 3-2 shows the block diagram of multi-unit optimization for two units. Both inputs are perturbed by Δ/2 and -Δ/2.

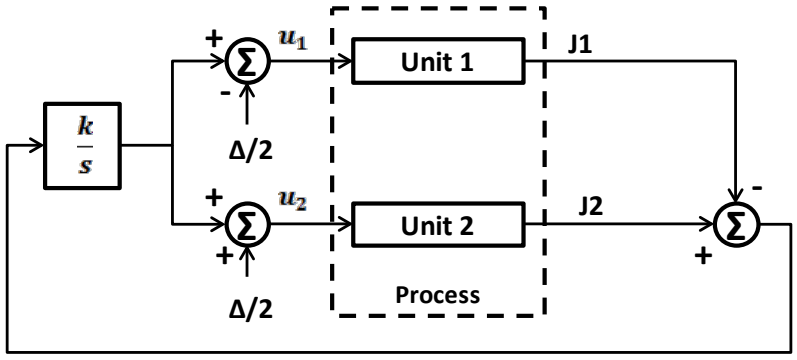

Fig 3-2: Schematic for multi-unit optimization for two units (Woodward et al. 2009a)

The optimization problem is to maximize a convex objective function with one input. A quadratic objective function is chosen as follow $\min_u J(u)$ in which

$$J(u) = (u - 2)^2 + 3 \qquad \text{Eq 9}$$

The first and second units are operated at the input values u1 and u2 respectively in which $u_1 = u - \frac{\Delta}{2}$, and $u_2 = u + \frac{\Delta}{2}$.

Then the gradient and the control law are given by:

$$\dot{u} = K\hat{g}(u) \qquad \text{Eq 10}$$

$$\hat{g}(u) = \frac{J_1(u_1) - J_2(u_2)}{\Delta} \qquad \text{Eq 11}$$

The results of applying multi-unit algorithm for two identical units with objective function as in Eq 9 is presented in Table 3-1. It is clear that $u^{opt} = 2$ and $J^{opt} = 3$. The letter N in the table means that the algorithm could not converge to the optimum point in run 7.

Table 3-1: summary of applying multi-unit without correctors on two identical units

Run	K	Δ	$\mathbf{u_0}$	u^*	u_1^*	u_2^*	J_1^*	J_2^*	**final $\hat{g}(u)$**
1	-100	0.25	-1	2	1.875	2.125	3.0156	3.0156	-8.5635e-005
2	-100	0.5	-1	2	1.75	2.25	3.0625	3.0625	-8.5635e-005
3	-100	1	-1	2	1.5	2.5	3.25	3.25	-8.5635e-005
4	-100	2	-1	2	1	3	4.0001	3.9999	-8.5635e-005
5	-1	0.25	-1	2	1.875	2.125	3.0156	3.0156	-9.9909e-005
6	-100	0.25	5	2	1.875	2.125	3.0156	3.0156	8.5635e-005
7	-1000	0.25	-1	N	N	N	N	N	N

u^* is the equilibrium point where the multi-unit algorithm converges so the inputs of the two units would converge to $u_1^* = u^* - \frac{\Delta}{2}$, and $u_2^* = u^* + \frac{\Delta}{2}$. In other word, both units inputs keep an offset Δ from each other. So choosing the offset Δ has an important role to make the algorithm converges to a circle around the optimum. If Δ is chosen too large, the algorithm converges to the optimum point but the assurance of converging to the optimum value of J is not obtained. Among different runs in the Table 3-1, run 4 shows the results with the biggest Δ.

K or the gain of integrator is another parameter which has to be chosen properly. If there is no dynamics in the system, K should be a value in which the algorithm converges to the optimum but $|K|$ should be determined in a way that there is not much moving on the static curve. If $|K|$ is too big, it means that it does not let the multi-unit converges or it may converge to a point which is far from optimum. It is because of the problem arises in making the system discrete. In other words it is numerical instability rather than control law instability. In run 7 an example of this fact is seen.

If the system has dynamics, K should not be faster than the system's dynamics. Besides these two parameters (Δ and K), the initial point for u in the algorithm should be set in the algorithm. In the case of identical units if there is priori knowledge about the system

optimal point, the initial point for u should be selected not so far from the optimum input. In other words, if the initial point for u is so far from the optimum input, it has effects on the algorithm's convergence, and the time needed for converging to the optimal point.

Fig 3-3, Fig 3-4, and Fig 3-5 show the Run 1, 5, and 6 respectively from Table 3-1. The unit of horizontal axis in all those graphs is the sample time of discrete system. Fig 3-4 shows the fact that if $|K|$ is chosen very small, it takes more time for the algorithm to converge to the optimum point. Comparing the results of run 5 and run 6 in Fig 3-5 displays that if the distance between u_0 and u^* are fixed, starting the algorithm from the right or left side of the optimum point on the static curve has no special effect on the convergence.

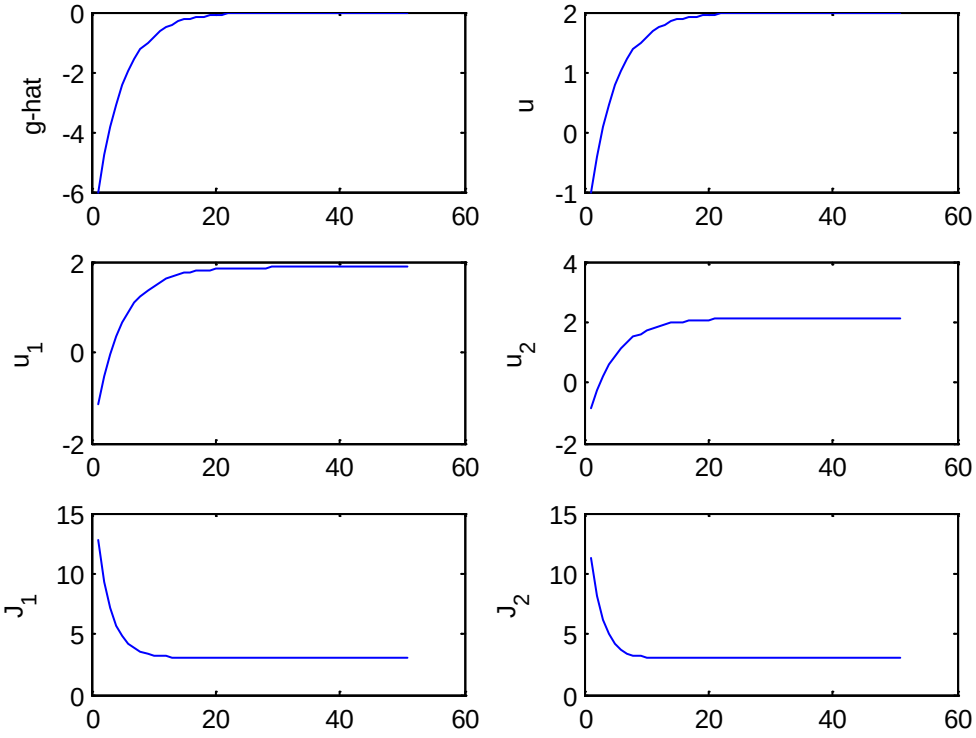

Fig 3-3: Multi-unit optimization for two identical units (Run 1)

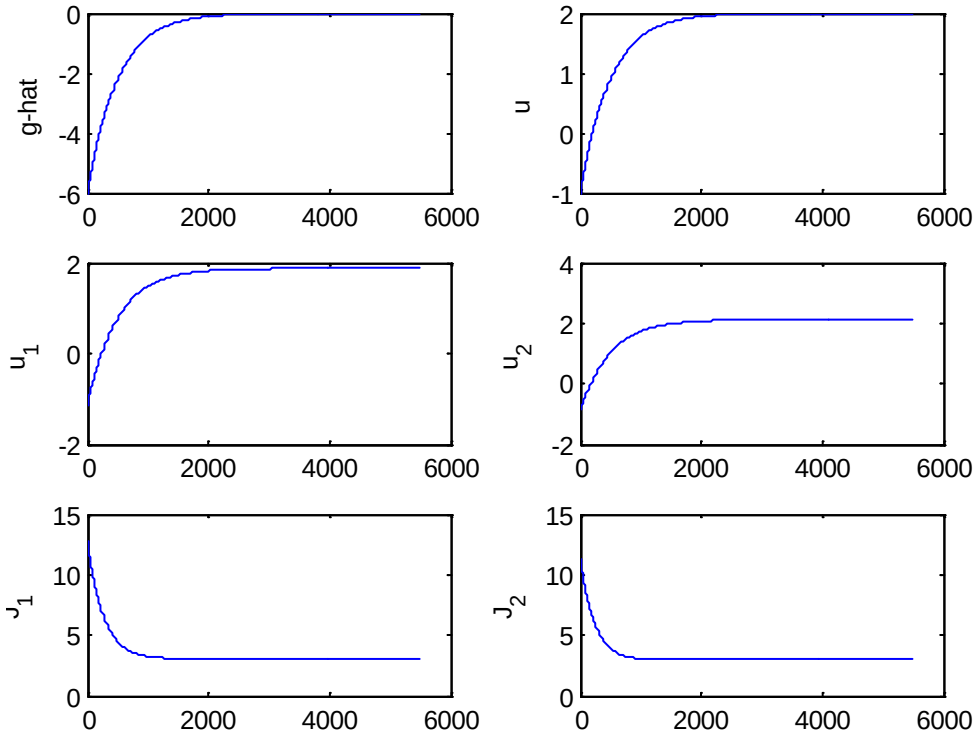

Fig 3-4: Multi-unit optimization for two identical units (Run 5)

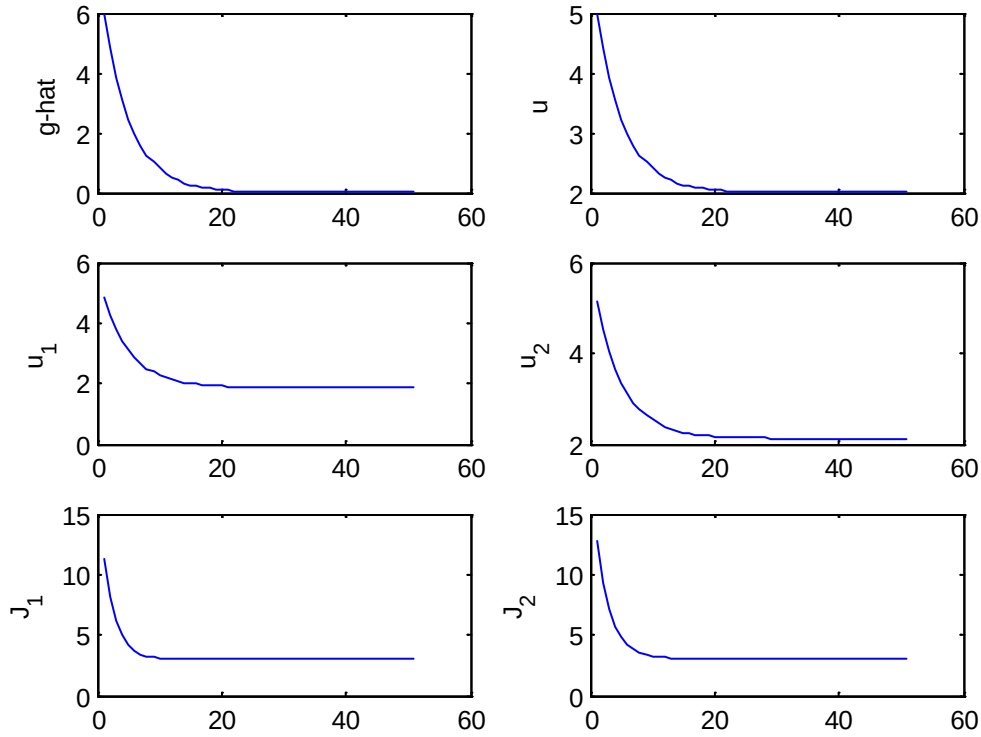

Fig 3-5: Multi-unit optimization for two identical units (Run 6)

3.4.2 Non-identical units

It was noted in research review that one of the main limitation of the multi-unit algorithm is the requirement of identical units. In this part based on what has been done by Woodward (Woodward et al. 2009a), the non-identical case is described. Then some simulation results are presented to express the correction phase effect more clearly.

3.4.2.1 Characterization of the differences between units

It is possible to establish difference between two units in different ways. For the non-identical case here, it is assumed that the dynamics are the same and very fast compared to the optimization time-scale so we have quasi-static process. Besides, the objective functions are convex and there is no noise effect considered. So the differences in units are from their static curves as it is shown in the Fig 3-6. In the other words, both curves have approximately similar shape.

In this figure, $\beta = u_1^{opt} - u_2^{opt}$ and $\lambda = J_2\left(u_2^{opt}\right) - J_1\left(u_1^{opt}\right)$. Besides, u_1^{opt} and u_2^{opt} are the optima of unit 1and 2 respectively.

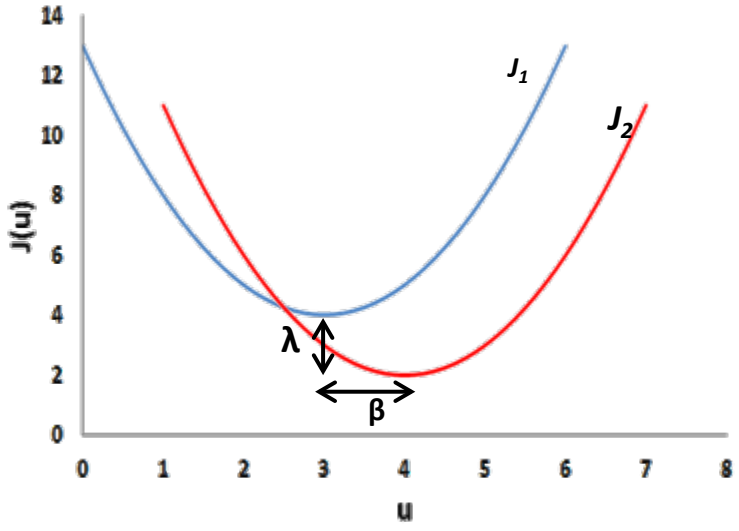

Fig 3-6: Difference between objective functions of two units

Both units follow the same control law and always keep an input offset of Δ from each other. Under these conditions, the static characteristics of the two units are represented using $J_1(u)$ and $J_2(u)$. The relationship between the two static maps is given by:

$$\boldsymbol{J_2(u) = J_1(u+\beta) + \lambda + \bar{J}(u+\beta)}$$ Eq 12

Because both static curves have smooth curvature at the optimum point it can be seen that:

$$\boldsymbol{\bar{J}\left(u_2^{opt}\right) = 0\ , \left.\frac{\partial \bar{J}}{\partial u}\right|_{u_2^{opt}} = 0}$$ Eq 13

So it is reasonable to assume that in the neighborhood of the optimum $\bar{J} = 0$. It was shown by Woodward that differences in the units cause the multi-unit scheme to a value away from the desired optimum; and the equilibrium point can be approximated by Eq 14 as follow:

$$u^* \approx \frac{u_1^{opt} + u_2^{opt}}{2} - \frac{\lambda}{(\Delta+\beta)\frac{\partial^2 J_1}{\partial u^2}}$$ Eq 14

It was also proved by Wood ward that the multi-unit algorithm for non-identical units is locally converge asymptotically if and only if the parameter Δ is chosen such that:

$$\Delta(\Delta + \beta) > 0$$ Eq 15

In the other words, convergence conditions can be satisfied in two different ways: by choosing the sign of Δ same as β or its absolute value is more than the absolute value of the β.

3.4.2.2 Multi-unit scheme with correction pattern

Adding a correction phase to the multi-unit phase, makes multi-unit algorithm converges to the optimal point. In this chapter the sequential correction approach is discussed. Correctors $\hat{\beta}$ and $\hat{\lambda}$ attenuate the effect of the differences in the optimal points of operation, and in the optimal values of the performance function respectively. By alternation between the multi-unit method and the calculation of correctors with two different perturbation signals they derived the update laws for the estimates (Woodward et al. 2009a; Woodward et al. 2009b). Perturbation signals for multi-unit with correctors are shown in Fig 3-7.

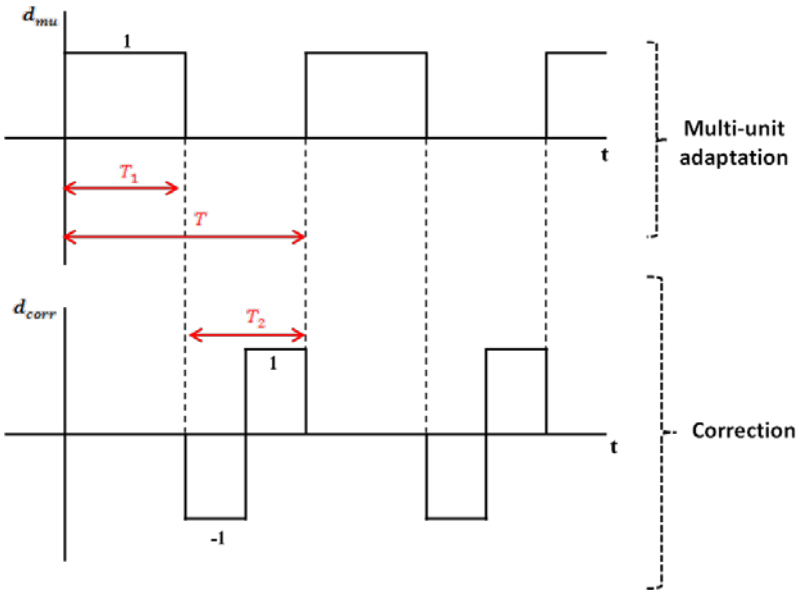

Fig 3-7: Perturbation signals for multi-unit with correctors (Woodward et al. 2009)

The structure of multi-unit algorithm with sequential correctors is displayed in Fig 3-8. In the correction phase, the difference between the two inputs, Δ, is removed. So, the two units act at the same operating point (corrected by $\hat{\beta}$ if any). Then, the corrected output values should be equal, if the vertical shift ($\hat{\lambda}$) is computed correctly. So, the difference between the corrected outputs provides the adaptation law for $\hat{\lambda}$.

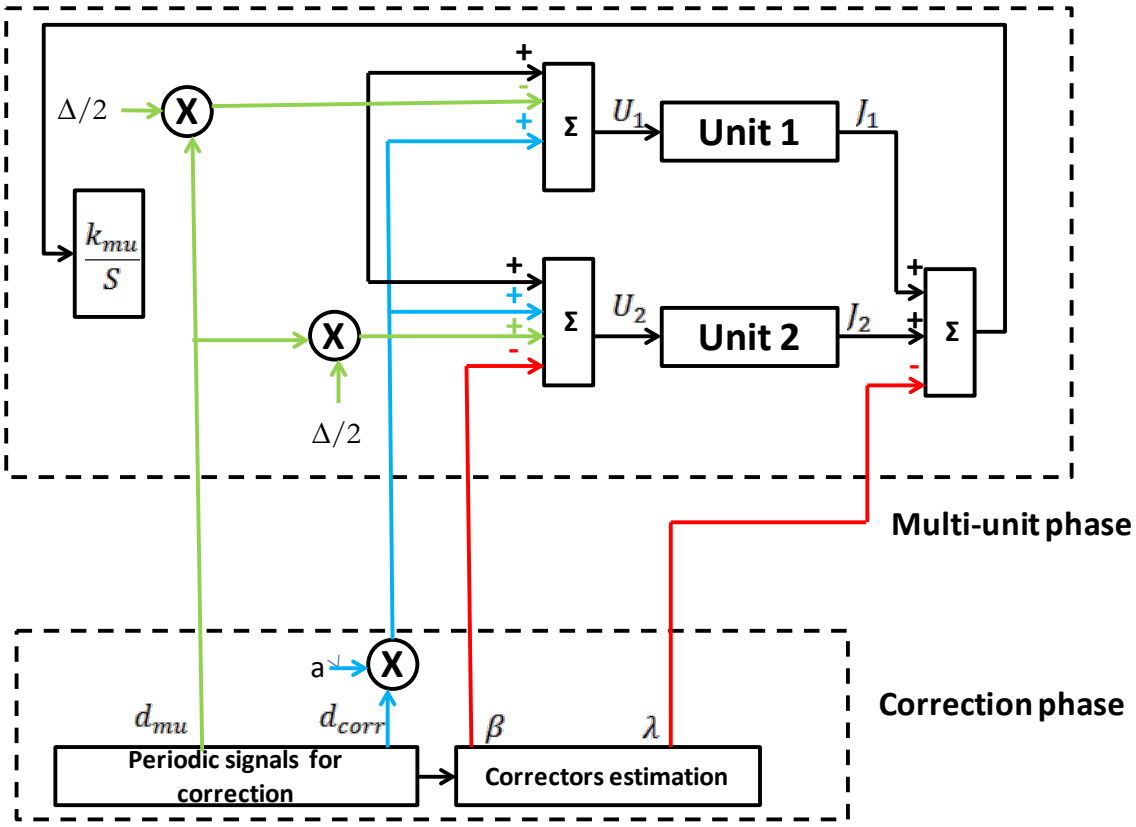

Fig 3-8: Structure of the multi-unit optimization method with sequential correctors

Synchronization of the inputs of two units is as following:

$$u_1 = u - \frac{\Delta}{2} d_{mu} + a d_{corr} \qquad \text{Eq 16}$$

$$u_2 = u + \frac{\Delta}{2} d_{mu} + a d_{corr} - \hat{\beta} \qquad \text{Eq 17}$$

Besides, the adaptation law for multi-unit is modified by this equation in which k_{mu} is the gain of integrator in multi-unit scheme.

$$\dot{u} = \frac{k_{mu}}{\Delta} (J_2 - J_1 - \hat{\lambda}) d_{mu} \qquad \text{Eq 18}$$

Moreover, two correctors are updated base on these adaptation laws:

$$\dot{\hat{\beta}} = \frac{k_\beta}{a} (J_2 - J_1 - \hat{\lambda}) d_{corr} \qquad \text{Eq 19}$$

$$\dot{\hat{\lambda}} = k_\lambda\left(J_2 - J_1 - \hat{\lambda}\right)(1 - d_{mu}) \qquad \text{Eq 20}$$

k_β and k_λ are the gains for correctors. Choosing the sign and value of these two are crucial in a general objective function. One way is to choose the starting points or u_0 for each unit in the algorithm in a such manner to have $\beta_0 = 0$. Then based on the shape of both static curves and the position of β_0 and λ_0 in relation to the correct values of β and λ for the optimal points, the sign of each gain could be determined. We can derive the following equation based on Eq 19:

$$\hat{\beta}(k+1) \approx \frac{k_\beta d_{corr}}{a}\left(J_2 - J_1 - \hat{\lambda}\right) + \hat{\beta}(k) \qquad \text{Eq 21}$$

If $\beta_0 < \beta$, the first statement in Eq 21 should be positive or in other words $\frac{\mathrm{k_\beta d_{corr}}}{\mathrm{a}}\left(\mathrm{J_2} - \mathrm{J_1} - \hat{\lambda}\right) > 0$. We know that the sign of a and $\mathrm{d_{corr}}$ is positive. Then by looking at the static curve we can estimate the sign of $\left(J_2 - J_1 - \hat{\lambda}\right)$ and this information helps to choose the correct sign for k_β. The same procedure would be helpful to know about the correct sign of k_λ.

There is no exact method which can justify how to choose the value for these corrector gains. It is more intuitive and based on trial and error.

a is the amplitude of perturbation or correction signal which should be fixed in the algorithm. T_1 and T_2 which are the periods for multi-unit and correction phase should be chosen relatively in such a way that the multi-unit phase has enough time to perform. When the objective function has dynamics the ratio of these periods ($\frac{T_1}{T_2}$) has an important role in convergence of multi-unit to the optimum point.

3.4.2.3 Simulation results and discussion for a generic case

To see the importance of correction phase multi-unit algorithm without correction is applied on two non-identical units with one input. First of all the condition for locally asymptotically convergence or the Eq 15 is considered. Assume two objective functions as follow should be maximized:

$$J_1 = (u-2)^2 + 3 \qquad \text{Eq 22}$$

$$J_2 = (u-6)^2 + 4 \qquad \text{Eq 23}$$

So $u_1^{opt} = 2$, $u_2^{opt} = 6$, $J_1^{opt} = 3$, $J_2^{opt} = 4$. As a result the correctors are $\beta = -4$ and $\lambda = 1$. Based on Eq 15, the algorithm would be stable if $\Delta < 0$ or $\Delta > 4$. To give a better insight of the condition, two different values of Δ are chosen and the multi-unit algorithm for identical unit is applied for the case of non-identical unit. Run 1 is with $\Delta = 0.25$ so $\Delta(\Delta + \beta) < 0$, and run 2 is with $\Delta = 5$ so $\Delta(\Delta + \beta) > 0$.

In Fig 3-9 the results of run 1 are shown. Both outputs of units are increasing and the simple algorithm without correction phase is diverging as it is deducted from the condition for guarantee the stability of the algorithm. Fig 3-10 displays the results of run 2. The algorithm is stable and both outputs of the units are converging but not to the optimal values. The unit of horizontal axis in all those graphs is the sample time of discrete system.

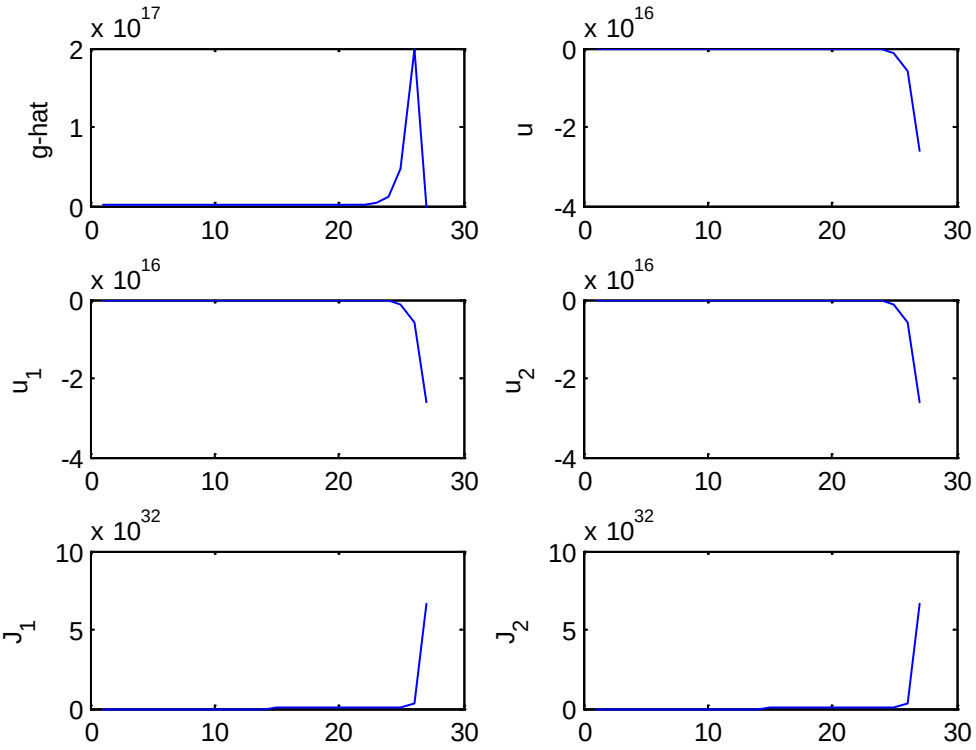

Fig 3-9: Multi-unit optimization without correction for two non-identical units (run 1)

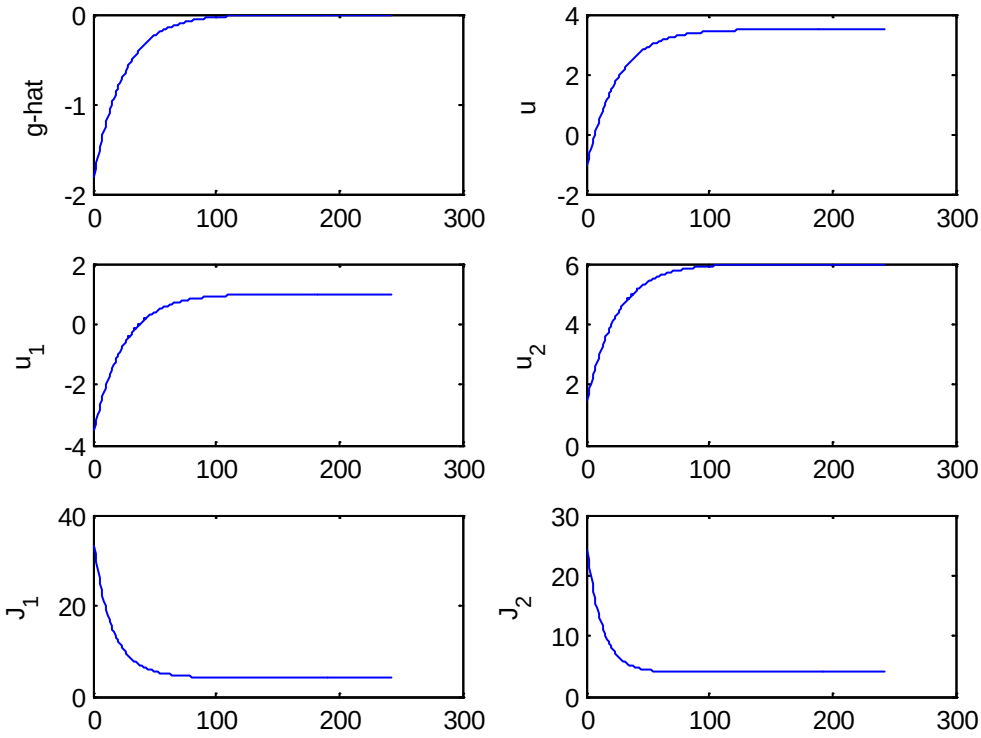

Fig 3-10: Multi-unit optimization without correction for two non-identical units (run 2)

A summary of the parameters and results for both runs are presented in Table 3-2. If the multi-unit works properly it is expected that:

$u_1^* = u_1^{opt} - \frac{\Delta}{2}$ Eq 24

$u_2^* = u_2^{opt} + \frac{\Delta}{2}$ Eq 25

In run1 the multi-unit scheme diverges because the offset parameter Δ is not in the range which is necessary to guarantee the stability of the algorithm. In run 2, the multi-unit scheme converges but not to the expected optimal point which are calculated by Eq 24 and Eq 25, as follow:

$u_1^* = 2 - 2.5 = -0.5$

$u_2^* = 6 + 2.5 = 8.5$

And based on Eq 14 the equilibrium point is:

$u^* = \frac{u_1^{opt} + u_2^{opt}}{2} - \frac{\lambda}{(\Delta+\beta)\frac{d^2J}{du^2}} = \frac{2+6}{2} - \frac{1}{(5-4)\times 2} = 3.5$

Table 3-2: summary of applying multi-unit without correctors on two non-identical units

Run	K	Δ	$\mathbf{u_0}$	u^*	u_1^*	u_2^*	$\mathbf{J_1}$	$\mathbf{J_2}$	$\hat{\mathbf{g}}^*(\mathbf{u})$
1	-100	0.25	-1	N	N	N	N	N	N
2	-100	5	-1	3.4998	0.9998	5.9998	4.0005	4.0000	-9.6081e-005

Now the proposed correction is applied on two non-identical units with objective functions similar to Eq 22, and Eq 23. Based on Eq 15, convergence conditions can be satisfied without corrections by $\text{sign}(\Delta) = \text{sign}(\beta)$ or $|\Delta| > |\beta|$. In the following simulations both values of $\Delta = 5$ and $\Delta = -2$ are used. The initial point is assumed equal to 4 so the initial correctors are as below:

$$u_0 = 4 \Longrightarrow \begin{cases} J_1 = (4-2)^2 + 3 = 7 \\ J_2 = (4-6)^2 + 4 = 8 \end{cases} \Longrightarrow \begin{cases} \beta_0 = 4 - 4 = 0 \\ \lambda_0 = 8 - 7 = 1 \end{cases}$$

The parameters of two runs are shown in Table 3-3. In Table 3-4, the results of using correction phase with multi-unit phase is presented. In both runs, the inputs and outputs of the two units are converging to the expected optimal points and optimal values. Besides, the correctors are converging to their ideal values.

Table 3-3: summary of parameters in applying multi-unit with correctors on two non-identical units

Run	$\mathbf{k_{mu}}$	Δ	$\mathbf{a}$	$\mathbf{k_\beta}$	$\mathbf{k_\lambda}$	$\mathbf{T_1}$	$\mathbf{T_2}$
1	-0.04	-2	0.5	0.01	0.015	100	100
2	-0.04	5	0.5	0.01	0.015	100	100

Table 3-4: summary of results in applying multi-unit with correctors on two non-identical units

Run	u^*	u_1^*	u_2^*	J_1^*	J_2^*	$\hat{\beta}^*$	$\hat{\lambda}^*$
1	2	3	5	4	5	-4	1
2	2	-0.5	8.5	9.25	10.25	-4	1

Although in all runs the inputs, outputs, and correctors converge to their expected values, the results presented for run 1 is more appropriate because $|\Delta|$ is smaller so the optimum converging values are nearer to the real optimum values. Fig 3-11 shows the graphs for input signals and the corrector $\hat{\beta}$ for the run1 of Table 3-4. The graphs for the corrector $\hat{\lambda}$, output signals, and correction signal in run 1 are displayed in Fig 3-12. Fig 3-13 and Fig 3-14 also presented the same results for the run 2 of Table 3-4. The unit of horizontal axis in all those graphs is the sample time of discrete system

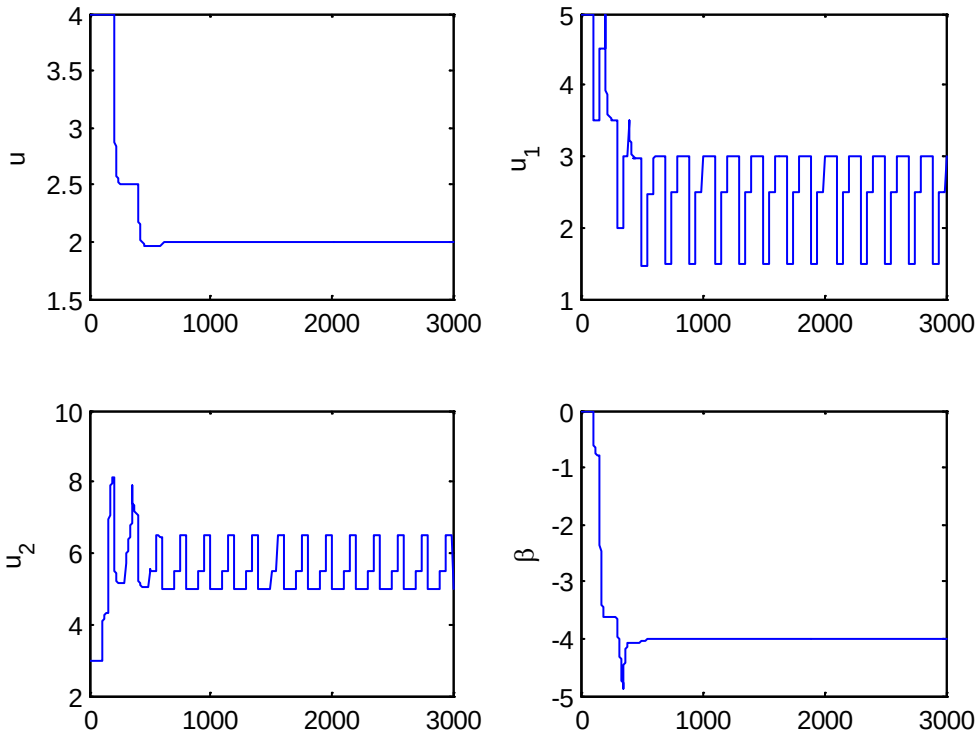

Fig 3-11: Input signals and corrector $\widehat{\boldsymbol{\beta}}$ in multi-unit scheme with correction for two non-identical units (run 1)

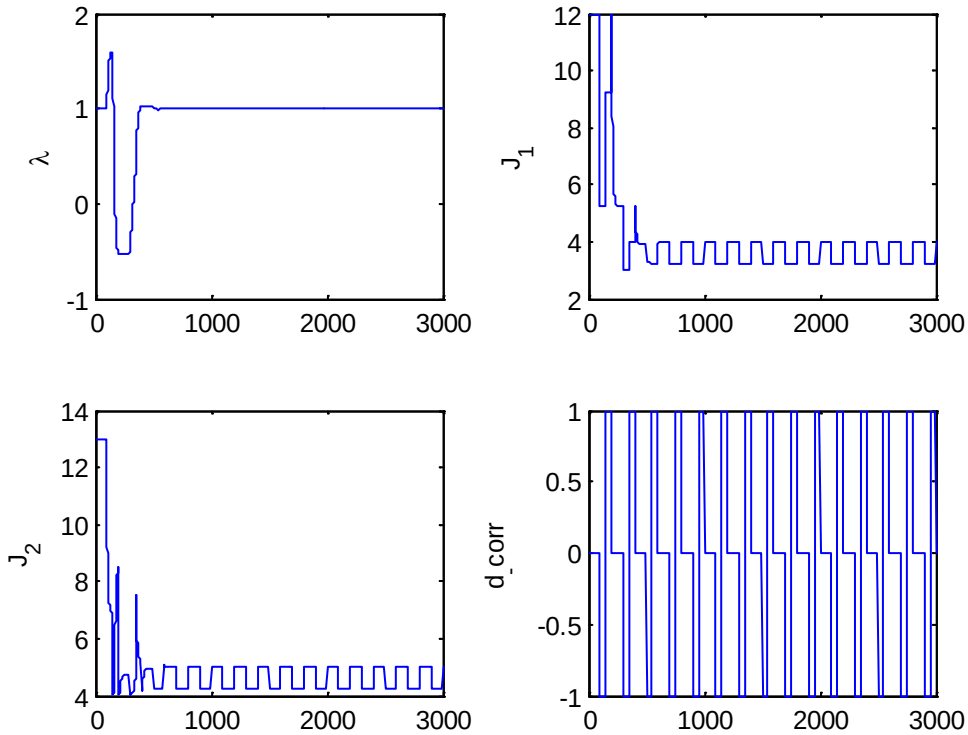

Fig 3-12: Output signals, corrector $\hat{\boldsymbol{\lambda}}$ and correction signal in multi-unit scheme with correction for two non-identical units (run 1)

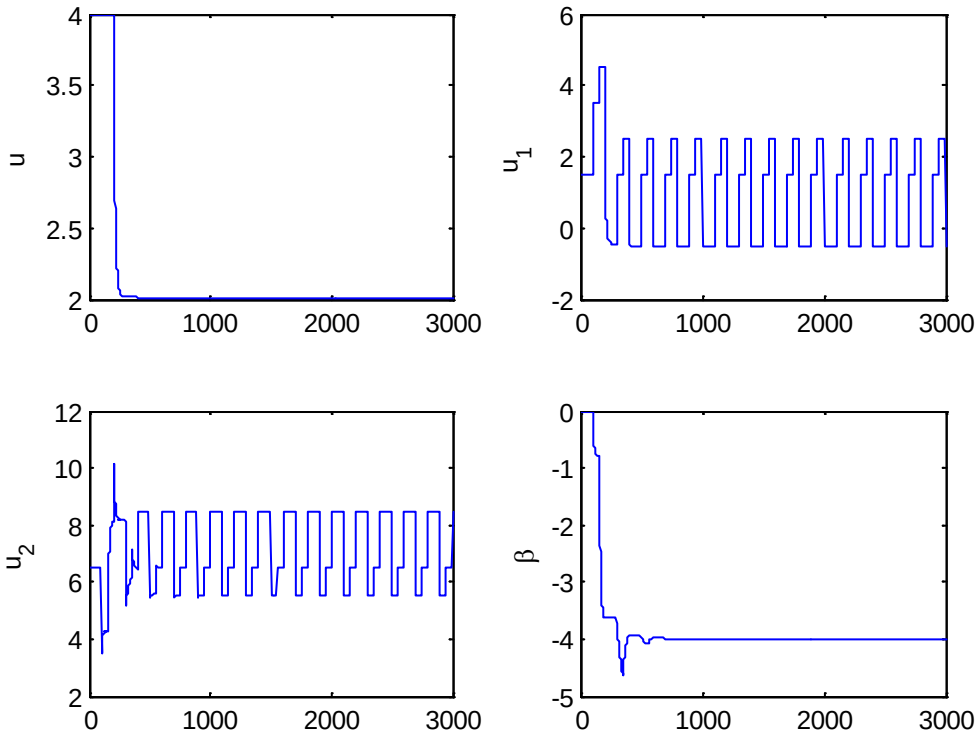

Fig 3-13: Input signals and corrector $\widehat{\boldsymbol{\beta}}$ in multi-unit scheme with correction for two non-identical units (run 2)

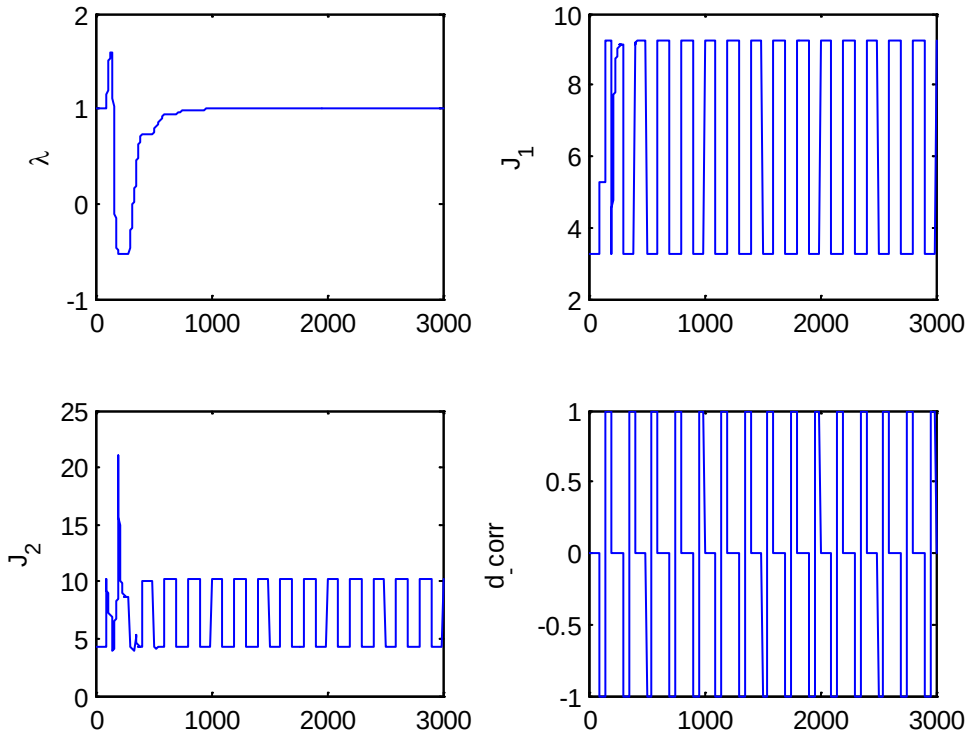

Fig 3-14: Output signals, corrector $\hat{\boldsymbol{\lambda}}$, and correction signal in multi-unit scheme with correction for two non-identical units (run 2)

3.5 Multi-unit optimization for three units and two inputs

The multi-unit optimization is discussed for the case of two units in previous part. Following what have been discussed, in this part the multi-unit optimization for three units and two inputs is investigated.

3.5.1 Identical units

Figure 3-15 shows the block diagram of multi-unit optimization for three units with two inputs. Input 1 of the unit two is perturbed by Δ and input 2 of the unit three is perturbed by Δ as well. Then the gradient is calculated by Eq 11, based on the differences between the outputs of units one and two, and the outputs of units one and three.

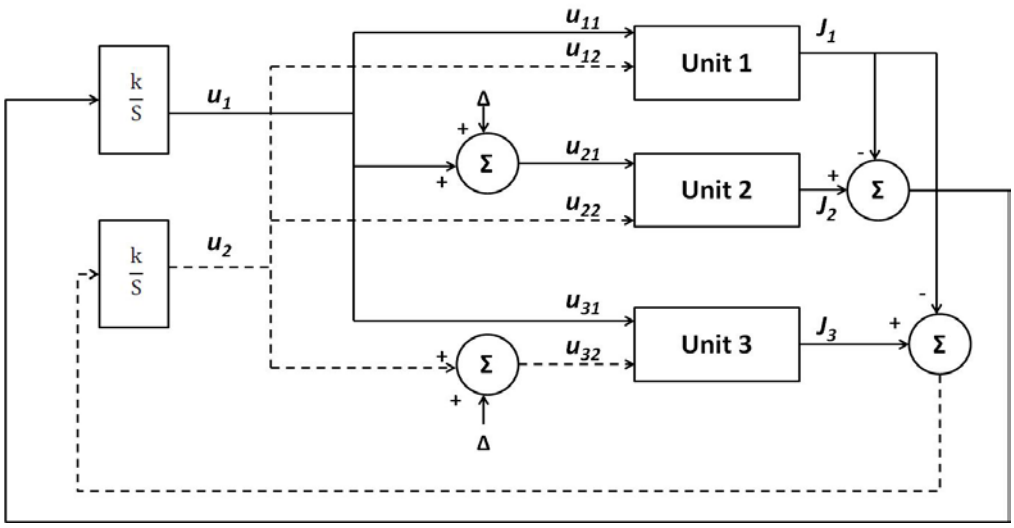

Fig 3-15: Schematic for multi-unit optimization for three units

The optimization problem is to maximize a convex objective function with two inputs. A quadratic objective function is chosen as follow $\min_u J(u)$ in which

$$J = (u_1 - 2)^2 + (u_2 - 3)^2 + 4 \qquad \text{Eq 26}$$

So $U^{opt} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ and $J^{opt} = 4$. The gradient and control law are given by Eq 10 and Eq 11. In other words, the negative sign for integrator is inside its gain. The results of applying multi-unit algorithm for three identical units with objective function as in Eq 26 is

presented in Table 3-5. Five runs are presented to show the effect of changing the offset Δ and gain of the integrator. In all runs, the initial point is assumed as $U_0 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$.

Table 3-5: Multi-unit optimization for three identical units and two inputs

Run	K	Δ	U^*	U_1^*	U_2^*	U_3^*	J_1^*	J_2^*	J_3^*	$\hat{\mathbf{g}}^*(\mathbf{u})$
1	-0.5	0.1	$\begin{bmatrix} 1.95 \\ 2.95 \end{bmatrix}$	$\begin{bmatrix} 1.95 \\ 2.95 \end{bmatrix}$	$\begin{bmatrix} 2.05 \\ 2.95 \end{bmatrix}$	$\begin{bmatrix} 1.95 \\ 3.05 \end{bmatrix}$	4.0050	4.0050	4.0050	$\approx \begin{bmatrix} 10^{-5} \\ 10^{-5} \end{bmatrix}$
2	-2	0.1	$\begin{bmatrix} 1.95 \\ 2.95 \end{bmatrix}$	$\begin{bmatrix} 1.95 \\ 2.95 \end{bmatrix}$	$\begin{bmatrix} 2.05 \\ 2.95 \end{bmatrix}$	$\begin{bmatrix} 1.95 \\ 3.05 \end{bmatrix}$	4.0050	4.0050	4.0050	$\approx \begin{bmatrix} 10^{-6} \\ 10^{-6} \end{bmatrix}$
3	-5	0.1	N	N	N	N	N	N	N	N
4	-2	0.2	$\begin{bmatrix} 1.9 \\ 2.9 \end{bmatrix}$	$\begin{bmatrix} 1.9 \\ 2.9 \end{bmatrix}$	$\begin{bmatrix} 2.1 \\ 2.9 \end{bmatrix}$	$\begin{bmatrix} 1.9 \\ 3.1 \end{bmatrix}$	4.02	4.02	4.02	$\approx \begin{bmatrix} 10^{-6} \\ 10^{-6} \end{bmatrix}$
5	-5	0.5	N	N	N	N	N	N	N	N

U^* shows the equilibrium point where the multi-unit algorithm converges for both inputs. The inputs of the three units would converge as following: $U_1^* = U^*$, $U_2^* = U^* + \begin{bmatrix} \Delta \\ 0 \end{bmatrix}$ and $U_3^* = U^* + \begin{bmatrix} 0 \\ \Delta \end{bmatrix}$. In other words, the Δ difference between input 1 of unit one and two, and between input 2 of unit one and three is remaining. K or the gain of the integrator has an effect on the convergence time; for example by increasing from 0.5 to 2 the time needed for convergence is decreasing from 80 samples to 30 samples. Besides, choosing the offset Δ has an important role similar to the two unit case.

In run 1, 2, and 4 all objective functions are fairly well equal to the optimal value which was 4. Finally gradient is small (approximately zero) which shows that the converging point is the optimal point. Fig 3-16 shows the inputs of each unit in run 2 from Table 3-5. In Fig 3-17, the input of the scheme, the output of the units, and the gradient are displayed for the run 2. The unit of horizontal axis in all those graphs is the sample time of discrete system.

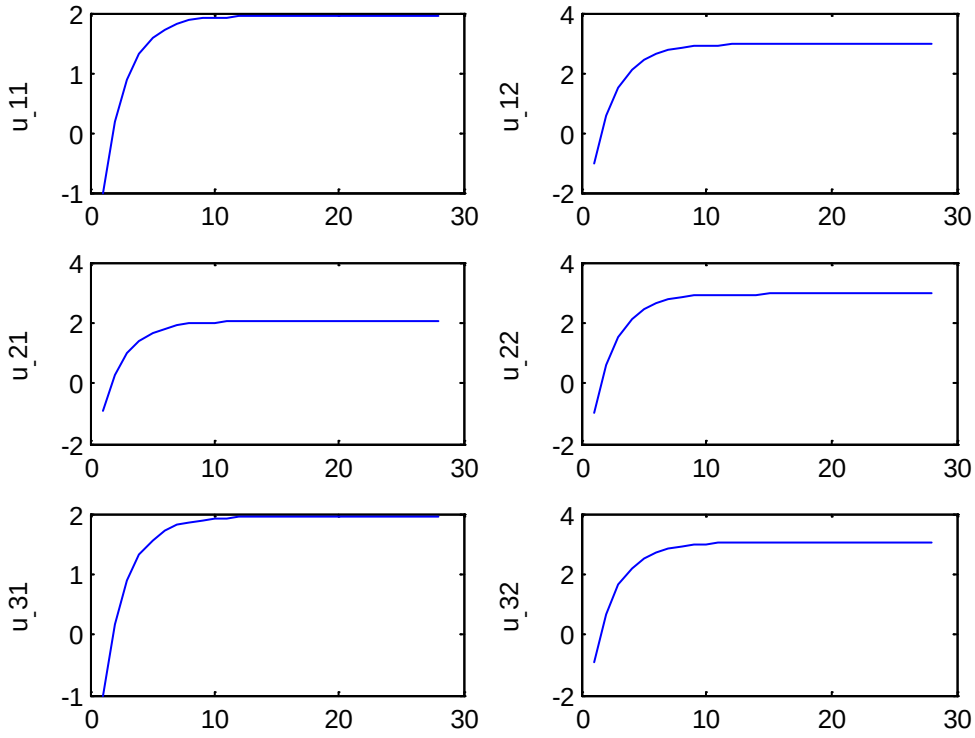

Fig 3-16: Input signals of unit in multi-unit optimization for three identical units

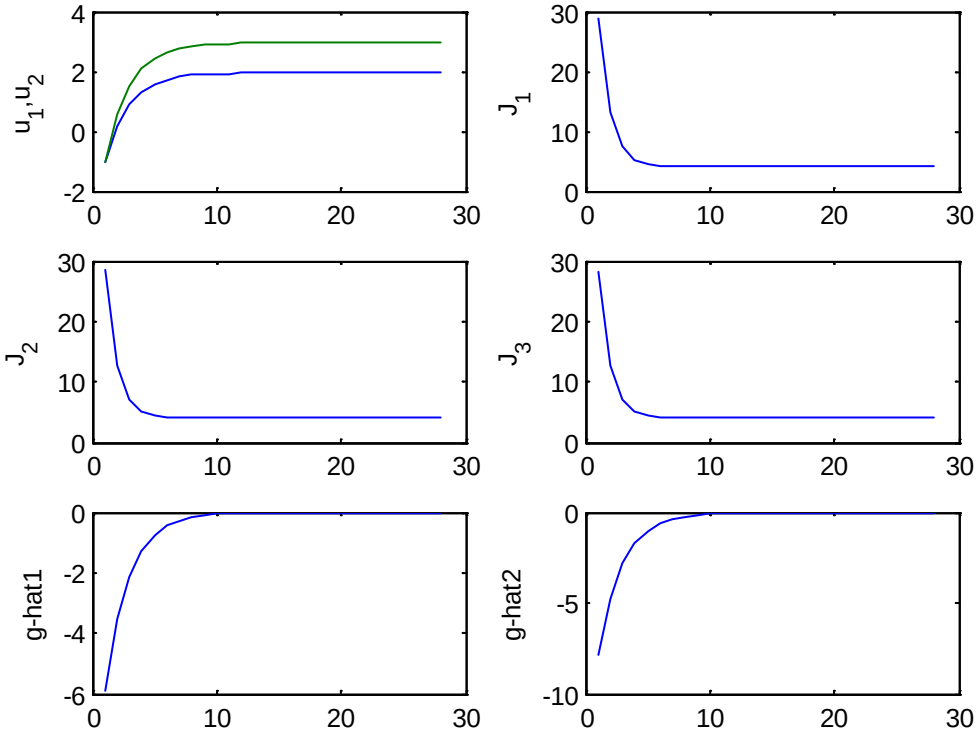

Fig 3-17: Input signal, Output of the units and gradient of unit in multi-unit optimization for three identical units

3.5.2 Non-identical units

3.5.2.1 Characterization of the differences between units

The difference between the units is between their static surfaces in three dimensions. Fig 3-18 shows this difference by an illustrative example.

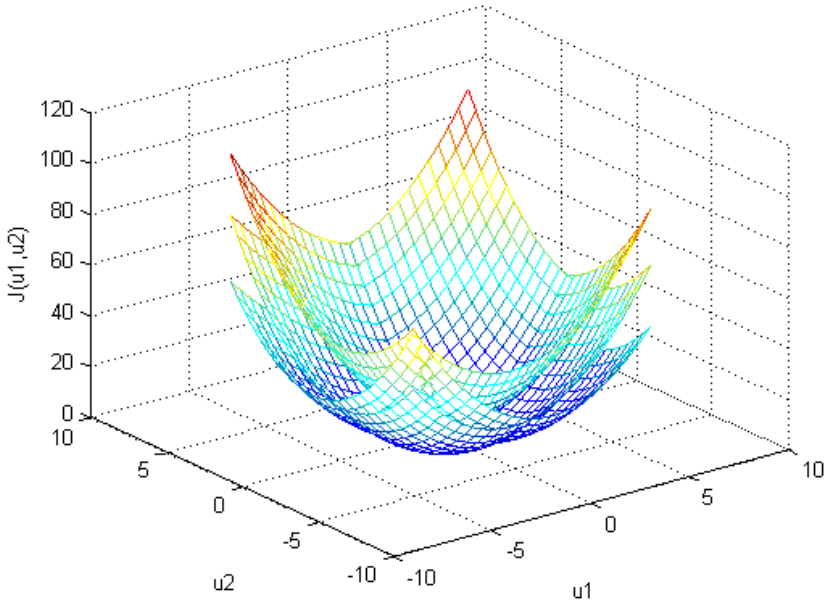

Fig 3-18: Differences between the static surfaces

If all the three objective functions are assumed to be convex. The relationships between the static surfaces of the units can be presented as follow:

$$J_2(U) = J_1(U + \beta) + \lambda + \bar{J}_{12}(U + \beta) \quad \text{Eq 27}$$

$$J_3(U) = J_1(U + \alpha) + \rho + \bar{J}_{13}(U + \alpha) \quad \text{Eq 28}$$

In which correctors β and α are two vectors, and λ and ρ are two scalars defined as:

$$U_1^{opt} - U_2^{opt} = \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \quad \text{Eq 29}$$

$$U_1^{opt} - U_3^{opt} = \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} \quad \text{Eq 30}$$

$$J_2^{opt} - J_1^{opt} = \lambda \quad \text{Eq 31}$$

$$J_3^{opt} - J_1^{opt} = \rho \qquad \text{Eq 32}$$

Because of the same smooth curvature at the optimal points, it can be derived that $\bar{J}_{12}(U_2^{opt}) = 0$, $\frac{\partial \bar{J}_{12}}{\partial U}\Big|_{U_2^{opt}} = 0$ and $\bar{J}_{13}(U_3^{opt}) = 0$, $\frac{\partial \bar{J}_{13}}{\partial U}\Big|_{U_3^{opt}} = 0$. So it is reasonable to assume that in the neighborhood of the optimums $\bar{J}_{12} = 0$ and $\bar{J}_{13} = 0$.

3.5.2.2 Multi-unit scheme with correction pattern

Based on the correction phase for the two non-identical units, in this part a schematic is proposed for three non-identical units and two inputs. In the Fig 3-19 the schematic of this developed multi-unit scheme is presented. Three different periodic signals are used to coordinate the static surfaces of three different objective functions. In Fig 3-19 $I = \begin{bmatrix}1\\1\end{bmatrix}$ and e_i is the i^{th} unit vector.

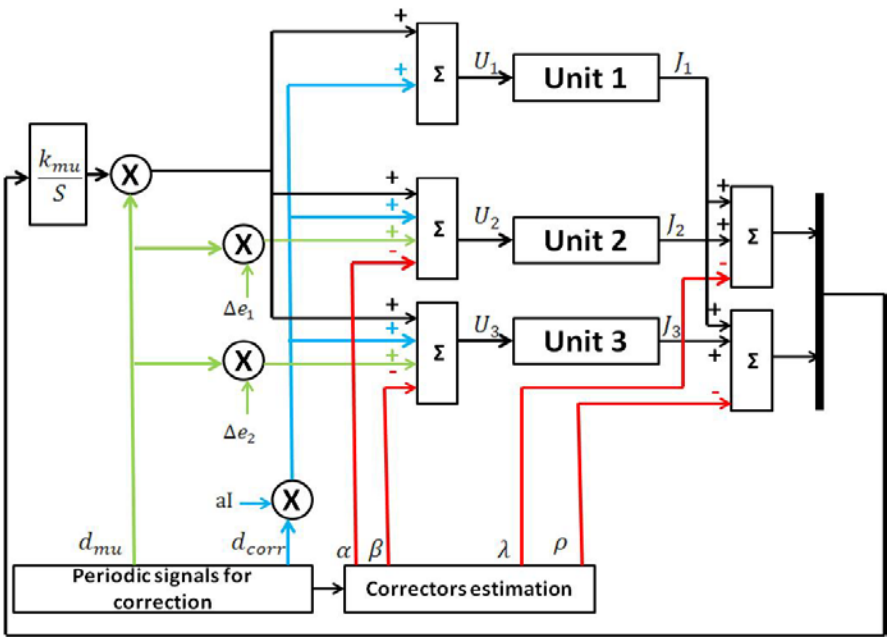

Fig 3-19: Structure of the multi-unit optimization method with sequential correctors for three units

Periodic signals for correction phase and multi-unit phase are different from what has been proposed for two units. We need to see the effect of perturbation to make correction between the output signals. Because the input is a 1×2 vector instead of a scalar, there is a need for two perturbation signals in the correction phase. In Fig 3-20 the perturbation

signals used for three non-identical units and two inputs are presented. Based on the multivariable extremum seeking scheme proposed by Ariyur and Krstic in 2002, the phase shift between these perturbation signals in correction phase is chosen $\frac{\pi}{2}$ (Ariyur and Krstic 2002). Therefore, the correction signals are orthogonal and defined as following:

$$d_{corr1} = sign\left(sin\left(\frac{2\pi t}{T_2}\right)\right) \quad \text{Eq 33}$$

$$d_{corr2} = sign\left(cos\left(\frac{2\pi t}{T_2}\right)\right) \quad \text{Eq 34}$$

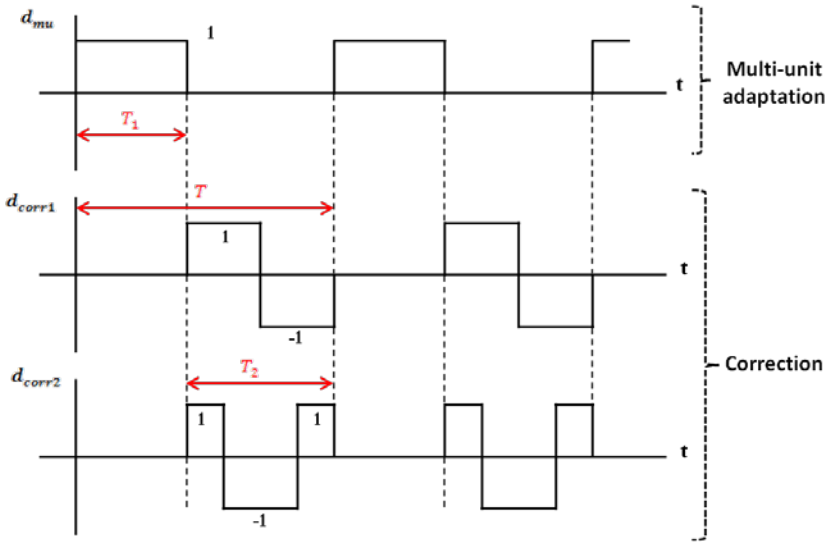

Fig 3-20: perturbation signals for multi-unit with correctors in the case of three non-identical units

In this scheme, synchronization of the input vectors of three units is as following

$$U_1 = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + ad_{corr1}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + ad_{corr2}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{Eq 35}$$

$$U_2 = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \Delta d_{mu}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + ad_{corr1}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + ad_{corr2}\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} \widehat{\beta_1} \\ \widehat{\beta_2} \end{bmatrix} \quad \text{Eq 36}$$

$$U_3 = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \Delta d_{mu}\begin{bmatrix} 0 \\ 1 \end{bmatrix} + ad_{corr1}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + ad_{corr2}\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} \widehat{\alpha_1} \\ \widehat{\alpha_2} \end{bmatrix} \quad \text{Eq 37}$$

Adaptation Laws for the input and correctors are proposed as follow:

$$\dot{U} = \frac{K_{mu} d_{mu}}{\Delta} \begin{bmatrix} J_2 - J_1 - \hat{\lambda} \\ J_3 - J_1 - \hat{\rho} \end{bmatrix} \quad \text{Eq 38}$$

$$\dot{\hat{\beta}} = \begin{bmatrix} \frac{k_{\beta 1} d_{corr1}}{a} \\ \frac{k_{\beta 2} d_{corr2}}{a} \end{bmatrix} (J_2 - J_1 - \hat{\lambda}) \quad \text{Eq 39}$$

$$\dot{\hat{\alpha}} = \begin{bmatrix} \frac{k_{\alpha 1} d_{corr1}}{a} \\ \frac{k_{\alpha 2} d_{corr2}}{a} \end{bmatrix} (J_3 - J_1 - \hat{\rho}) \quad \text{Eq 40}$$

$$\dot{\hat{\lambda}} = k_{\lambda}(1 - d_{mu})(J_2 - J_1 - \hat{\lambda}) \quad \text{Eq 41}$$

$$\dot{\hat{\rho}} = k_{\rho}(1 - d_{mu})(J_3 - J_1 - \hat{\rho}) \quad \text{Eq 42}$$

In the proposed scheme for three non-identical units, choosing the sign and value of the gains of correctors is more complicated than the case with two non-identical units. Like before, first of all the parameters Δ , a , and the multi-unit gain, d_{mu}, should be chosen properly. Though the signs of the gains of correctors can be preset based on the position of static surfaces related to the objective functions of the units, the proper values of the gains of correctors are extracted based on trial and error.

3.5.2.3 Simulation results and discussion for a generic case

To see the importance of correction phase, multi-unit algorithm without correction is applied on three non-identical units with two inputs. Assume three objective functions as follow should be minimized:

$$J_1 = (u_1 - 2)^2 + (u_2 - 3)^2 + 4 \quad \text{Eq 43}$$

$$J_2 = (u_1 - 1)^2 + (u_2 - 4)^2 + 6 \quad \text{Eq 44}$$

$$J_3 = (u_1 - 3)^2 + (u_2 - 5)^2 + 3 \quad \text{Eq 45}$$

So $U_1^{opt} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$, $U_2^{opt} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$, $U_3^{opt} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$, $J_1^{opt} = 4$, $J_2^{opt} = 6$, $J_3^{opt} = 3$. As a result the correctors are $\beta = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, $\alpha = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$, $\lambda = 2$, and $\rho = -1$.

In the following simulation two different cases are assumed for the parameter Δ. In run 1 it is assumed that $|\Delta| < |\beta|$, and $|\Delta| < |\alpha|$. Vise versa, in run 2 it is assumed that $|\Delta| > |\beta|$, and $|\Delta| > |\alpha|$. Assumptions for both runs are as below:

1. run1: $U_0 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$, $\Delta = 0.1$, and $K = 0.5$
2. run 2: $U_0 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$, $\Delta = 3$, and $K = 0.5$

The results of applying multi-unit algorithm without correctors for run 1 are shown in Fig 3-21, and Fig 3-22. And the results of applying multi-unit algorithm without correctors for run 2 are shown in Fig 3-23, and Fig 3-24. Both outputs of units are increasing and the simple algorithm without correction phase is diverging.

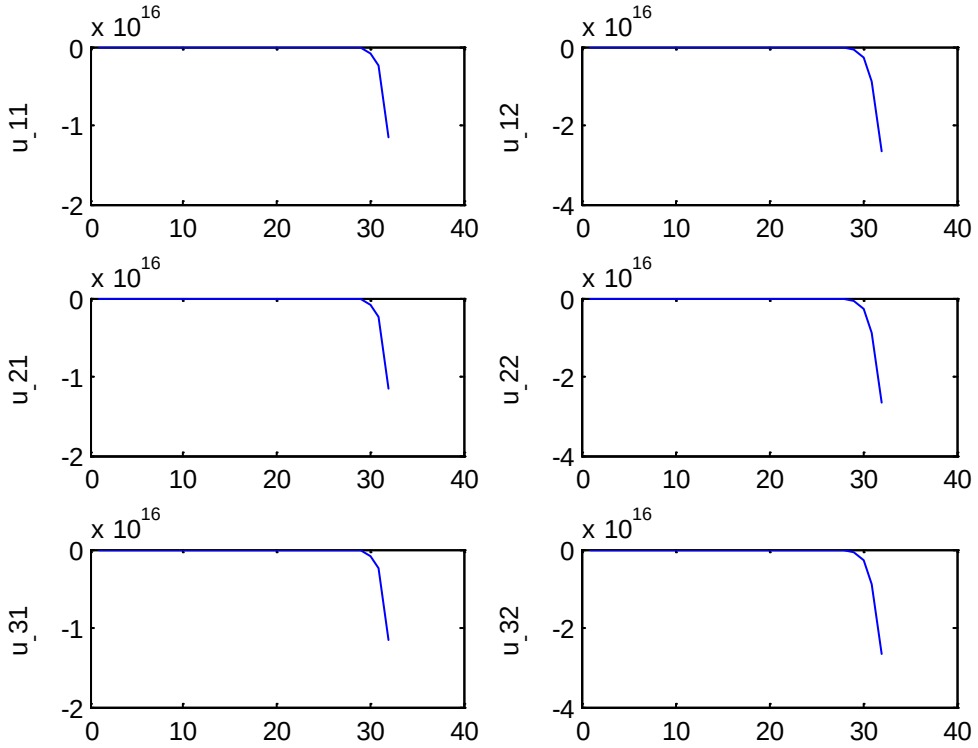

Fig 3-21: Input signals of units in multi-unit optimization for three non-identical without correction (run 1)

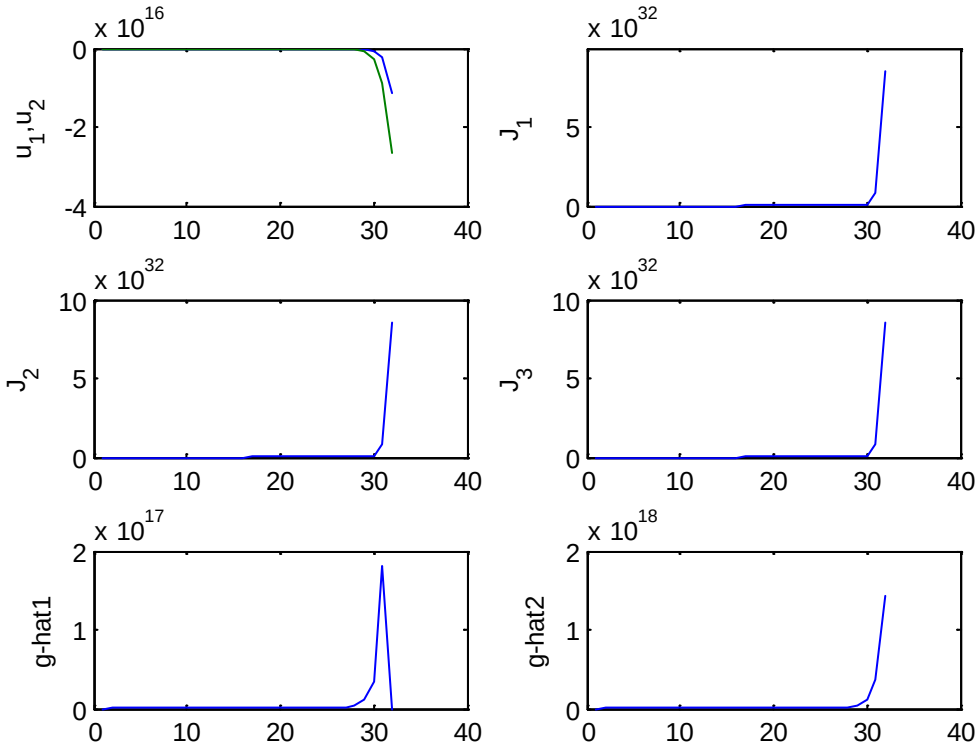

Fig 3-22: Input signal, Output of the units and gradient in multi-unit optimization for three non-identical without correction (run 1)

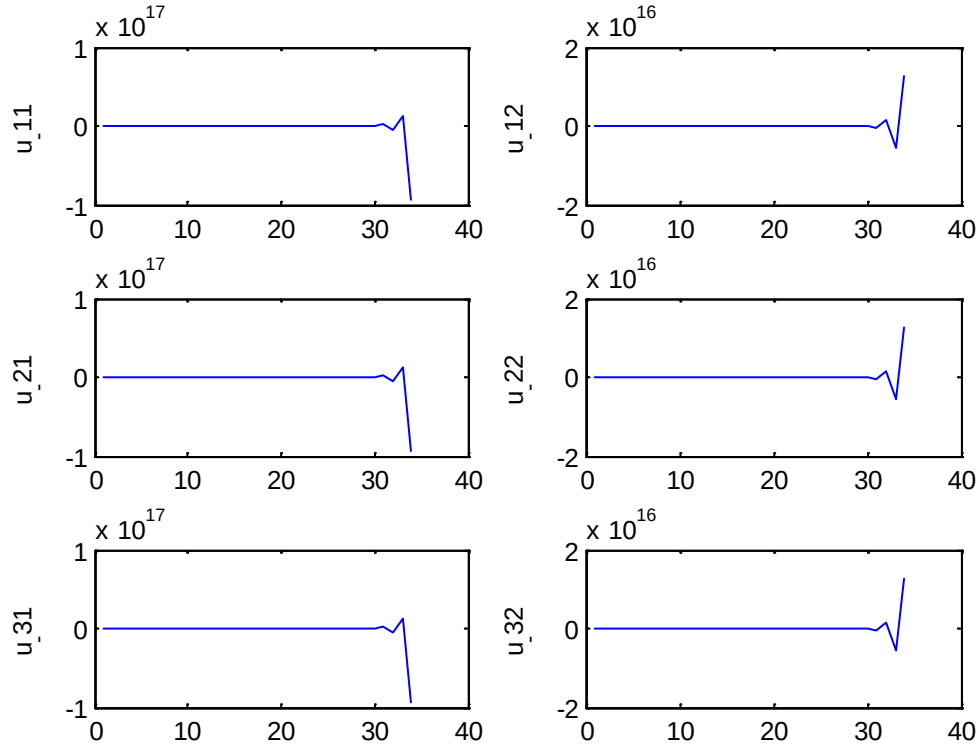

Fig 3-23: Input signals of units in multi-unit optimization for three non-identical without correction (run 2)

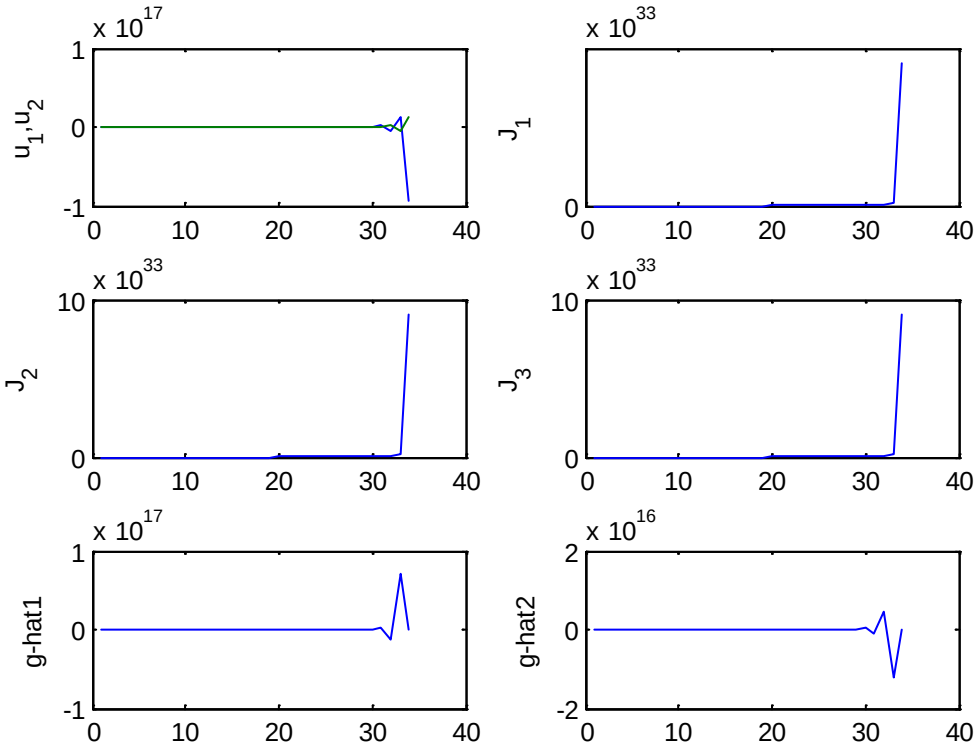

Fig 3-24: Input signal, Output of the units and gradient in multi-unit optimization for three non-identical without correction (run 2)

Now the proposed correction is applied on three non-identical units with objective functions similar to Eq 43, Eq 44 and Eq 45. In the following simulation two cases are assumed for the initial conditions:

1. $U_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Longrightarrow \begin{cases} J_1 = 4 + 9 + 4 = 17 \\ J_2 = 1 + 16 + 6 = 23 \\ J_3 = 9 + 25 + 3 = 37 \end{cases}$

So the initial values for correctors are $\beta_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\alpha_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\lambda_0 = 23 - 17 = 6$, and $\rho_0 = 37 - 17 = 20$.

2. $U_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Longrightarrow \begin{cases} J_1 = 1 + 4 + 4 = 9 \\ J_2 = 0 + 9 + 6 = 15 \\ J_3 = 4 + 16 + 3 = 23 \end{cases}$

So the initial values for correctors are $\beta_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\alpha_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\lambda_0 = 15 - 9 = 6$, and $\rho_0 = 23 - 9 = 14$.

Among the numerous runs, 16 runs are selected to show the importance of tuning the gains and parameters in multi-unit method with correctors for three non-identical units

and two inputs. First, a proper value for Δ is chosen based on the difference between the optimal points on static surfaces of the objective functions in each units. Then the parameter a is set. After that, the gains of multi-unit and correction phases are chosen with some trial and error. In all of these runs $T_1 = 100$, and $T_2 = 100$ is chosen for periods of perturbation signals.

In Table 3-6, the initial values for inputs and correctors in different runs are shown. The values of parameters in each run are displayed in Table 3-7. The parameter n shows the number of oscillations in each period for the two correcting perturbation signals. For example if $n = 4$ the perturbation signals would be the same as Fig 3-23.

Table 3-6: summary of initial values for inputs and correctors in multi-unit algorithm with correctors on three non-identical units

Run	U_0	β_0	α_0	λ_0	ρ_0
1	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
2	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
3	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
4	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
5	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
6	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
7	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
8	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
9	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
10	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
11	$\begin{bmatrix}1\\1\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	14
12	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
13	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
14	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
15	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20
16	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	6	20

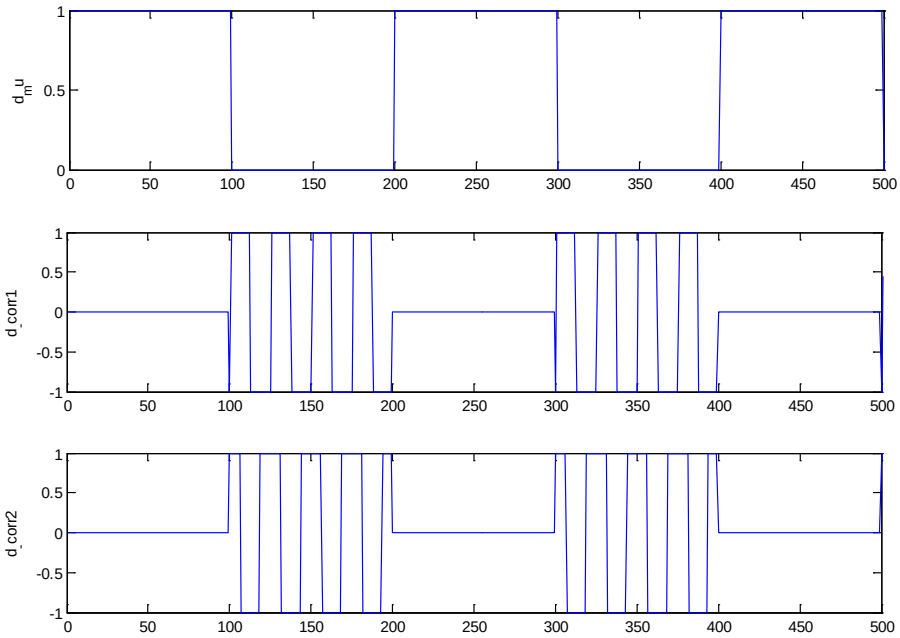

Fig 3-25: perturbation signals with n=4 oscillations in each period

Table 3-8 shows the optimal points and values, and the values of estimated correctors by applying the multi-unit with correctors for the 16 runs. The symbol N means that the algorithm does not converge in the relative run. The true values of the optimal points and values for each unit, and the true values of the correctors are displayed in Table 3-9 to evaluate the functionality of multi-unit method.

Table 3-7: summary of parameters in multi-unit algorithm with correctors on three non-identical units

Run	$\mathbf{k_{mu}}$	Δ	**a**	$\mathbf{k_{\beta}}$	$\mathbf{k_{\alpha}}$	$\mathbf{k_{\lambda}}$	$\mathbf{k_{\rho}}$	**n**
1	-0.05	3	1	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	0.1	1
2	-0.05	2	0.9	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	0.1	1
3	-0.005	2	0.9	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	0.1	1
4	-0.005	0.9	0.4	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	0.1	1
5	-0.001	0.9	0.4	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	0.1	1
6	-0.005	0.9	0.4	$\begin{bmatrix}0.01\\0.01\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	0.1	1
7	-0.005	0.9	0.4	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.03\\0.01\end{bmatrix}$	0.01	0.1	1
8	-0.005	0.9	0.4	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.1	0.1	1
9	-0.005	0.9	0.4	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	1	1
10	-0.005	0.7	0.4	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	0.1	1
11	-0.005	0.7	0.4	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	0.1	1
12	-0.005	0.7	0.4	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	0.1	4
13	-0.005	0.6	0.4	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	0.1	1
14	-0.005	0.6	0.3	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	0.1	1
15	-0.001	0.6	0.3	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	0.1	1
16	-0.005	0.6	0.3	$\begin{bmatrix}0.001\\0.001\end{bmatrix}$	$\begin{bmatrix}0.003\\0.001\end{bmatrix}$	0.01	0.1	4

Table 3-8: summary of results in applying multi-unit with correctors on three non-identical units

Run	u_1^*	u_2^*	u_3^*	J_1	J_2	J_3	$\widehat{\beta}^*$	$\widehat{\alpha}^*$	$\hat{\lambda}^*$	$\widehat{\rho}^*$
1	$\begin{bmatrix}0.5093\\1.478\end{bmatrix}$	$\begin{bmatrix}2.5\\2.496\end{bmatrix}$	$\begin{bmatrix}1.539\\6.495\end{bmatrix}$	8.538	10.51	7.369	$\begin{bmatrix}1.009\\-1.018\end{bmatrix}$	$\begin{bmatrix}-1.03\\-2.017\end{bmatrix}$	1.978	-1.169
2	N	N	N	N	N	N	N	N	N	N
3	$\begin{bmatrix}0.9998\\2.01\end{bmatrix}$	$\begin{bmatrix}2\\3.005\end{bmatrix}$	$\begin{bmatrix}1.977\\6.017\end{bmatrix}$	5.98	7.989	5.079	$\begin{bmatrix}0.9998\\-0.9955\end{bmatrix}$	$\begin{bmatrix}-0.9791\\-2.007\end{bmatrix}$	2.009	-0.9159
4	$\begin{bmatrix}1.551\\2.562\end{bmatrix}$	$\begin{bmatrix}1.45\\3.559\end{bmatrix}$	$\begin{bmatrix}2.929\\5.465\end{bmatrix}$	4.394	6.399	3.439	$\begin{bmatrix}1.001\\-0.9971\end{bmatrix}$	$\begin{bmatrix}-0.9831\\-2.005\end{bmatrix}$	2.003	-0.9879
5	$\begin{bmatrix}1.528\\2.635\end{bmatrix}$	$\begin{bmatrix}1.429\\3.627\end{bmatrix}$	$\begin{bmatrix}2.492\\5.532\end{bmatrix}$	4.359	6.325	3.534	$\begin{bmatrix}0.9993\\-0.995\end{bmatrix}$	$\begin{bmatrix}-0.9722\\-1.999\end{bmatrix}$	2.003	-0.992
6	N	N	N	N	N	N	N	N	N	N
7	N	N	N	N	N	N	N	N	N	N
8	$\begin{bmatrix}1.725\\2.561\end{bmatrix}$	$\begin{bmatrix}1.486\\3.494\end{bmatrix}$	$\begin{bmatrix}2.702\\5.463\end{bmatrix}$	4.268	6.492	3.303	$\begin{bmatrix}-1.138\\-0.9394\end{bmatrix}$	$\begin{bmatrix}-0.9837\\-2.003\end{bmatrix}$	2.221	-0.9713
9	$\begin{bmatrix}1.559\\3.926\end{bmatrix}$	$\begin{bmatrix}1.453\\4.924\end{bmatrix}$	$\begin{bmatrix}1.014\\5.589\end{bmatrix}$	5.05	7.063	7.291	$\begin{bmatrix}1.005\\-1.001\end{bmatrix}$	$\begin{bmatrix}N\\-0.7632\end{bmatrix}$	2.01	2.252
10	$\begin{bmatrix}1.655\\2.651\end{bmatrix}$	$\begin{bmatrix}1.352\\3.651\end{bmatrix}$	$\begin{bmatrix}2.647\\5.363\end{bmatrix}$	4.242	6.563	3.255	$\begin{bmatrix}1.003\\-1\end{bmatrix}$	$\begin{bmatrix}-0.9936\\-2.012\end{bmatrix}$	2.002	-1.015
11	$\begin{bmatrix}1.655\\2.649\end{bmatrix}$	$\begin{bmatrix}1.352\\3.649\end{bmatrix}$	$\begin{bmatrix}2.651\\5.362\end{bmatrix}$	4.242	6.563	3.252	$\begin{bmatrix}1.004\\-1\end{bmatrix}$	$\begin{bmatrix}-0.997\\-2.012\end{bmatrix}$	2.004	-1.014
12	$\begin{bmatrix}1.65\\2.654\end{bmatrix}$	$\begin{bmatrix}1.35\\3.652\end{bmatrix}$	$\begin{bmatrix}2.648\\5.351\end{bmatrix}$	4.243	6.244	3.247	$\begin{bmatrix}0.9996\\-0.9984\end{bmatrix}$	$\begin{bmatrix}-0.9988\\-1.998\end{bmatrix}$	2.001	-0.9977
13	$\begin{bmatrix}1.707\\2.663\end{bmatrix}$	$\begin{bmatrix}1.302\\3.666\end{bmatrix}$	$\begin{bmatrix}2.742\\5.289\end{bmatrix}$	4.199	6.015	3.526	$\begin{bmatrix}1.004\\-1.002\end{bmatrix}$	$\begin{bmatrix}-1.035\\-2.025\end{bmatrix}$	2.001	-1.046
14	N	N	N	N	N	N	N	N	N	N
15	$\begin{bmatrix}1.673\\2.789\end{bmatrix}$	$\begin{bmatrix}1.273\\3.784\end{bmatrix}$	$\begin{bmatrix}2.631\\5.393\end{bmatrix}$	4.15	6.121	3.28	$\begin{bmatrix}0.9991\\-0.9953\end{bmatrix}$	$\begin{bmatrix}-0.966\\-1.996\end{bmatrix}$	2.001	-0.9569
16	$\begin{bmatrix}1.701\\2.702\end{bmatrix}$	$\begin{bmatrix}1.3\\3.702\end{bmatrix}$	$\begin{bmatrix}2.7\\5.301\end{bmatrix}$	4.178	6.179	3	$\begin{bmatrix}1.001\\-0.9992\end{bmatrix}$	$\begin{bmatrix}-0.9992\\-1.998\end{bmatrix}$	2.001	-0.9985

Table 3-9: summary of real values for the optimal points and correctors

u_1^*	u_2^*	u_3^*	J_1	J_2	J_3	β	α	λ	ρ
$\begin{bmatrix}2\\3\end{bmatrix}$	$\begin{bmatrix}1\\4\end{bmatrix}$	$\begin{bmatrix}3\\5\end{bmatrix}$	4	6	3	$\begin{bmatrix}1\\-1\end{bmatrix}$	$\begin{bmatrix}-1\\-2\end{bmatrix}$	2	-1

By comparing the results of run 1 and run 2 form Table 3-8, it can be realized that if the values of Δ and a are decreasing but the multi-unit gain does not change relatively, the algorithm does not converge. Therefore to recover convergence the multi-unit gain should be decreased by the factor of 5 as it is shown in run 3. However the equilibrium points in run 1 and run 3 do not necessarily equal to the optimal points for each unit.

Comparison between run 3 and run 4 shows that by decreasing the amount of multi-unit gain the algorithm converges to near optimal points.

It can be realized from run 4 and run 6 that if the gain of corrector β is increased by the factor of 10 the algorithm diverges. Besides from run 4 and run 7, it can be deduced that if the gain of corrector α is increased by the factor of 10 the algorithm diverges, too.

By comparing run 4 and run 8, it is realized that if the gain of corrector λ is increased by the factor of 10 the algorithm still converges to the near of optimal points.

Though increasing the gain of corrector ρ by the factor of 10, from run 4 to run 9, does not force the algorithm to diverge but it has significant impact on the convergence of α_1 and it leads the algorithm to converge not near the optimal points.

The graphs in Fig 3-26 to Fig 3-28 show the input signals, output signals, and the estimated correctors for run 10. The input signals, output signals, and the estimated correctors for run 12 are also displayed in Fig 3-29 to Fig 3-31. Comparing the results of run 10 and run 11 show that changing in initial inputs or the vector U_0, and as a result changing in initial guess for the correctors λ, and ρ has no significant impact on the convergence of the algorithm.

Results of run 10 and run 12 display that if more than one oscillation exist in one period of perturbation signals, such as Fig 3-25, the algorithm still converges to very near to optimal points. Furthermore, the convergence is faster in run 12 that in run 10.

Run 14 are related to a situation in which the multi-unit algorithm diverges and by some modifications in run 15 and run 16, it has been tried to make it converge. First by

decreasing the multi-unit gain by the factor of 5, the algorithm converges in run 15 which shows the improvement from run 14. After that, by using 4 oscillations in one period of perturbation signals in run 16, such as Fig 3-25, the algorithm converges to very near optimal points; it means adding more oscillations improve the multi-unit algorithm functionality from run 14 to run 16. The graphs in Fig 3-32 to Fig 3-34 show the input signals, output signals, and the estimated correctors for run 15. The input signals, output signals, and the estimated correctors for run 16 are also displayed in Fig 3-35 to Fig 3-37.

Another comparison between the results of run 10 in Fig 3-27, and the results of run 15 in Fig 3-33 shows that although the steady state response is converging faster in run 10 than in run 15, the inappropriate transient response in run 10 is modified by decreasing the multi-unit gain in run 15.

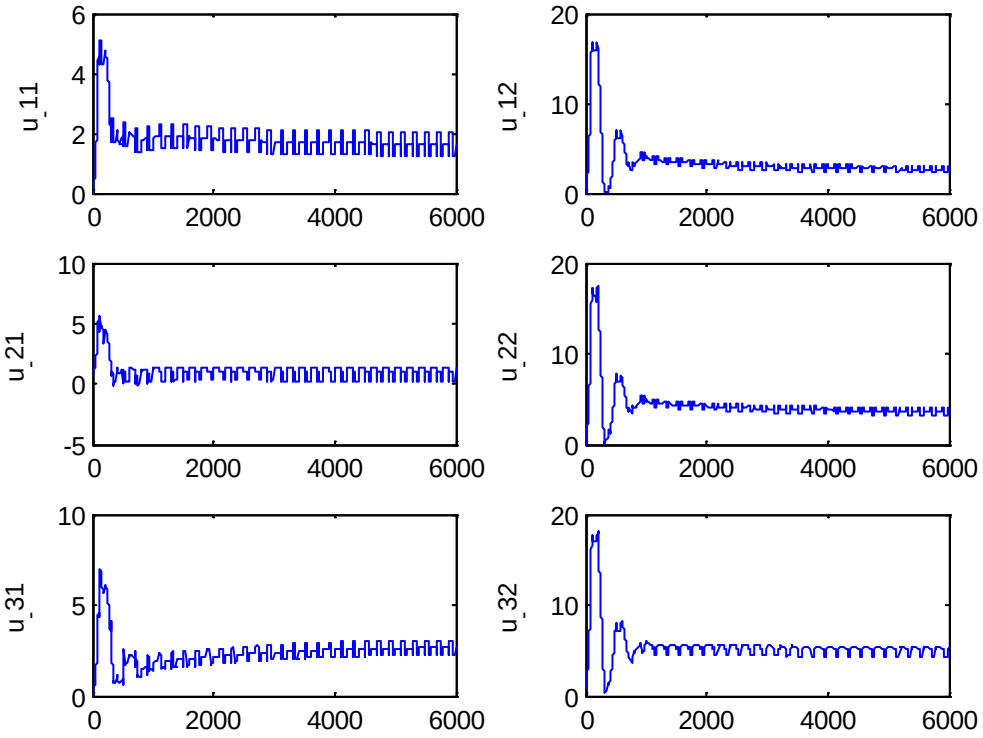

Fig 3-26: Input signals in multi-unit scheme with correction for three non-identical units (run 10)

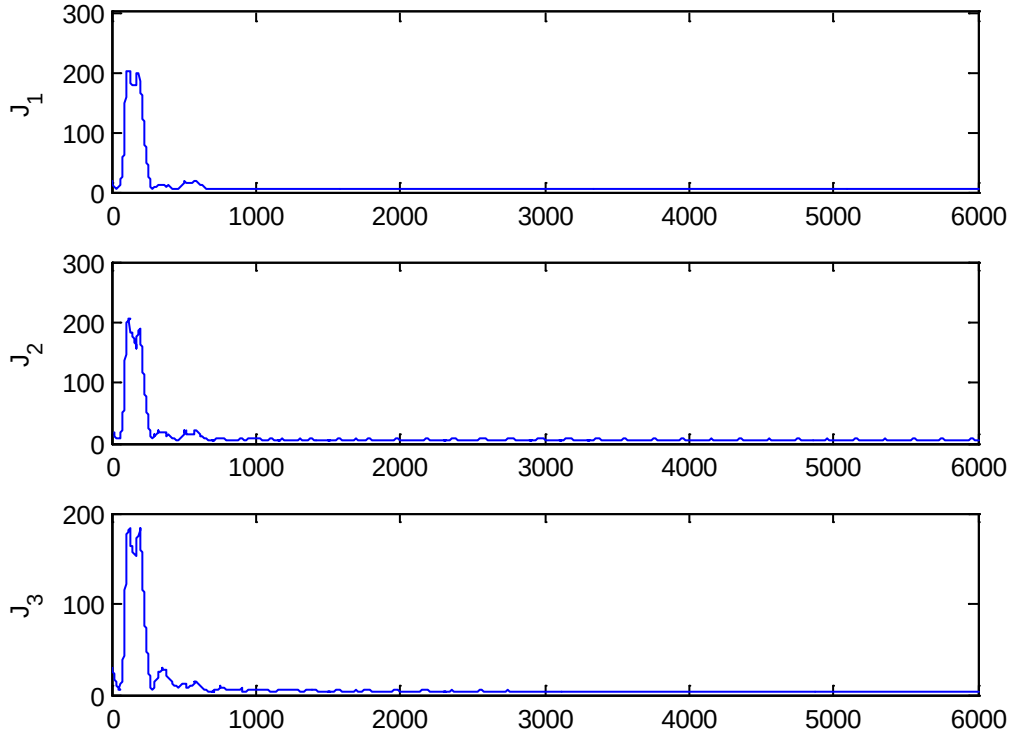

Fig 3-27: Output signals in multi-unit scheme with correction for three non-identical units (run 10)

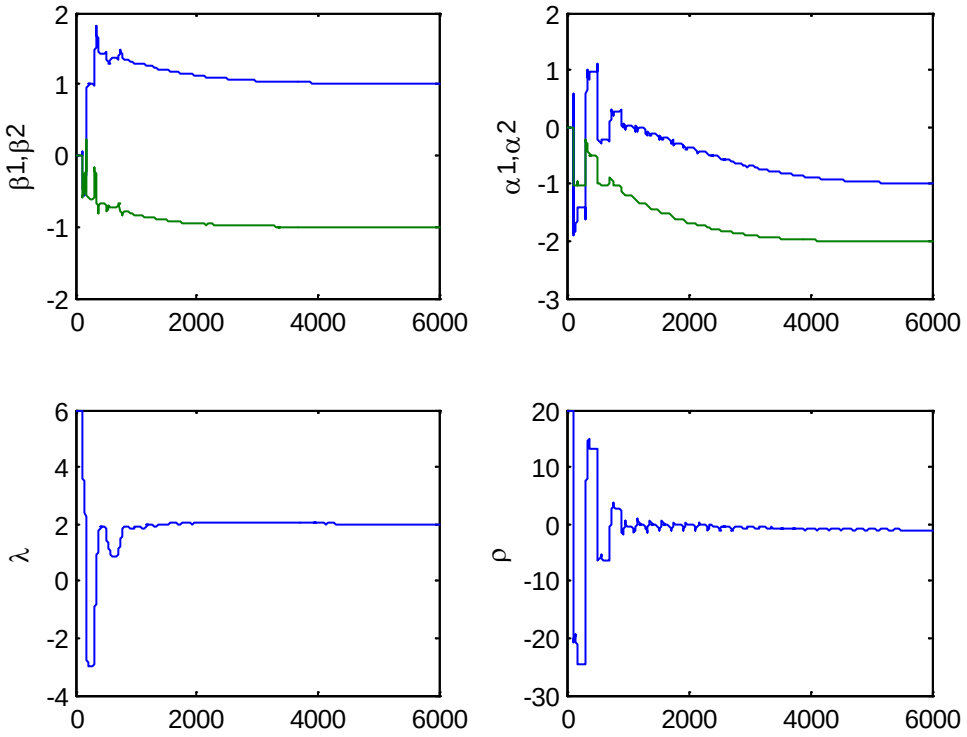

Fig 3-28: Estimated correctors in multi-unit scheme with correction for three non-identical units (run 10)

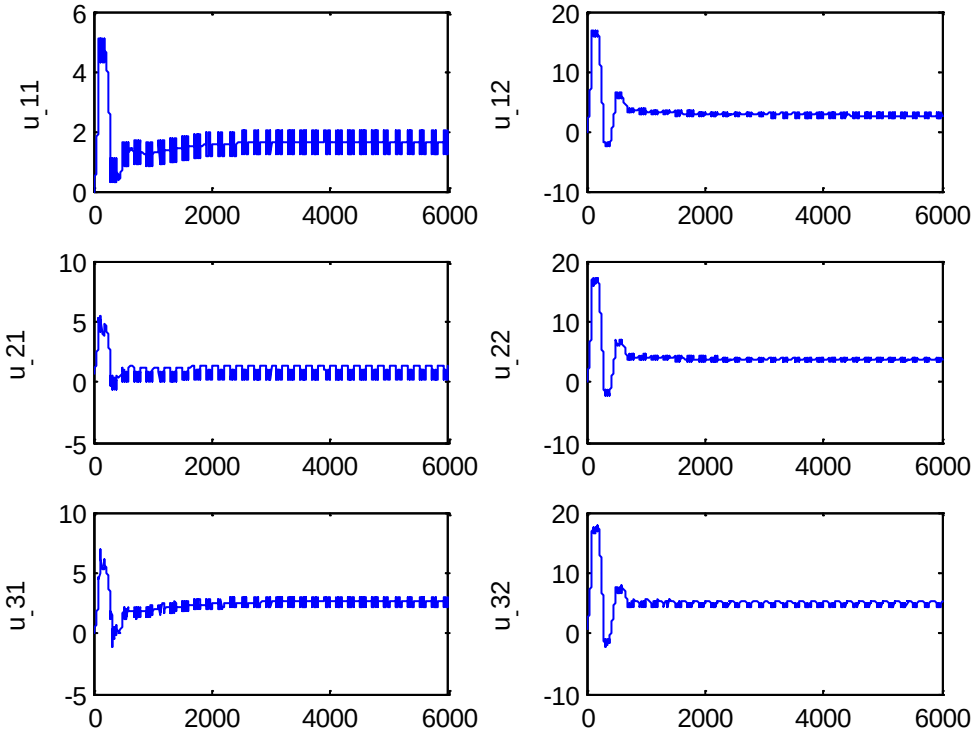

Fig 3-29: Input signals in multi-unit scheme with correction for three non-identical units (run 12)

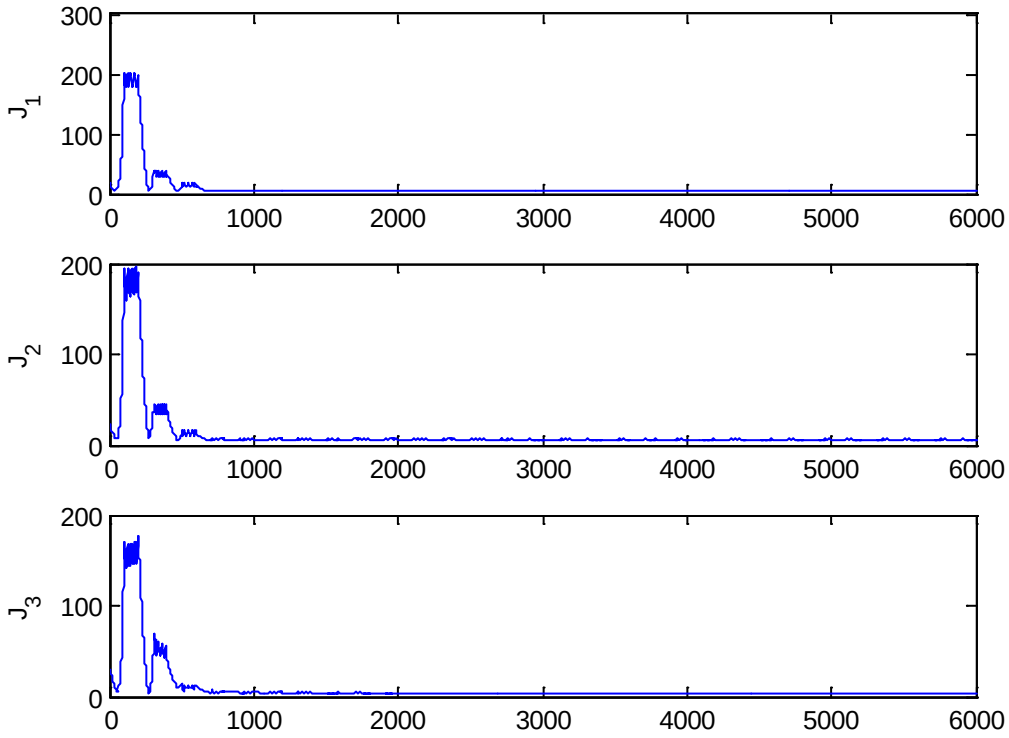

Fig 3-30: Output signals in multi-unit scheme with correction for three non-identical units (run 12)

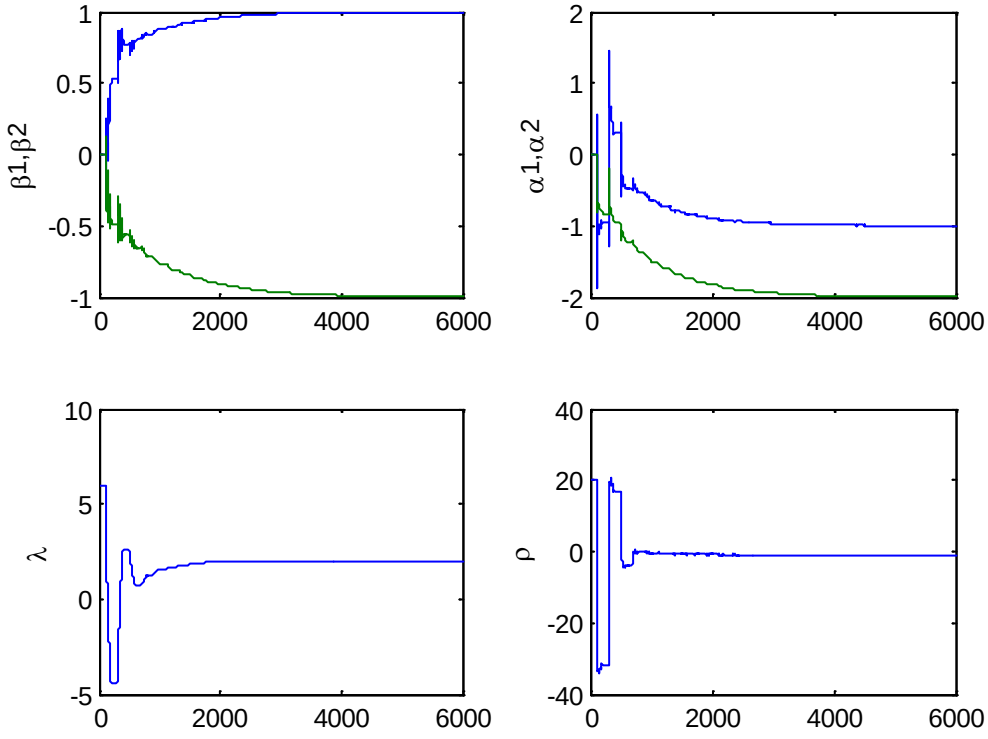

Fig 3-31: Estimated correctors in multi-unit scheme with correction for three non-identical units (run 12)

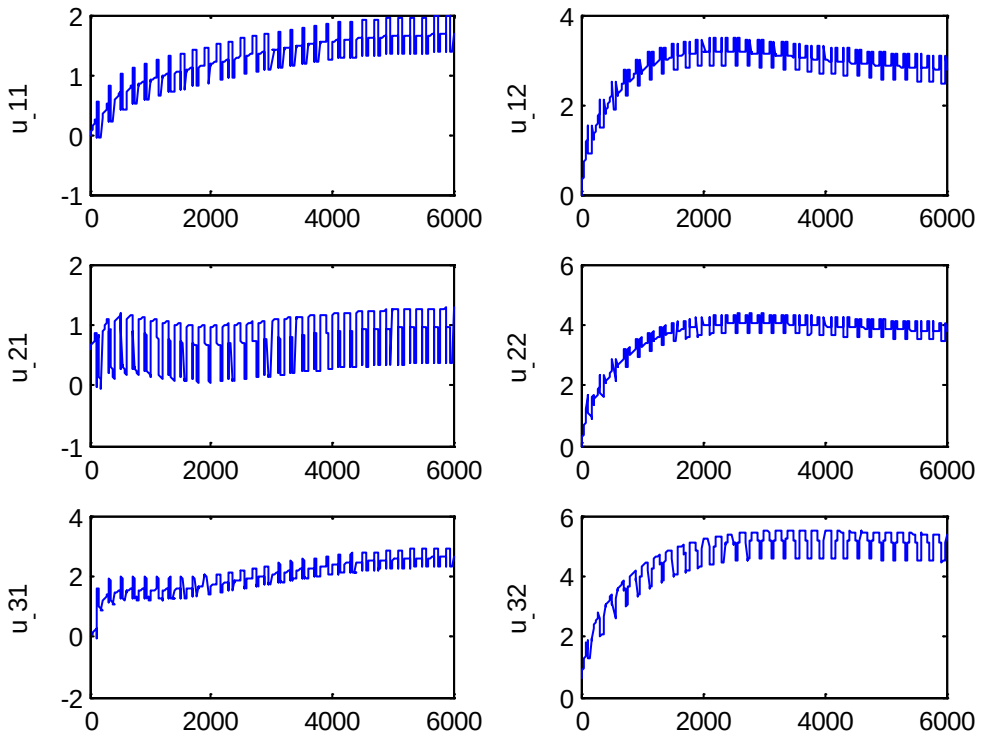

Fig 3-32: Input signals in multi-unit scheme with correction for three non-identical units (run 15)

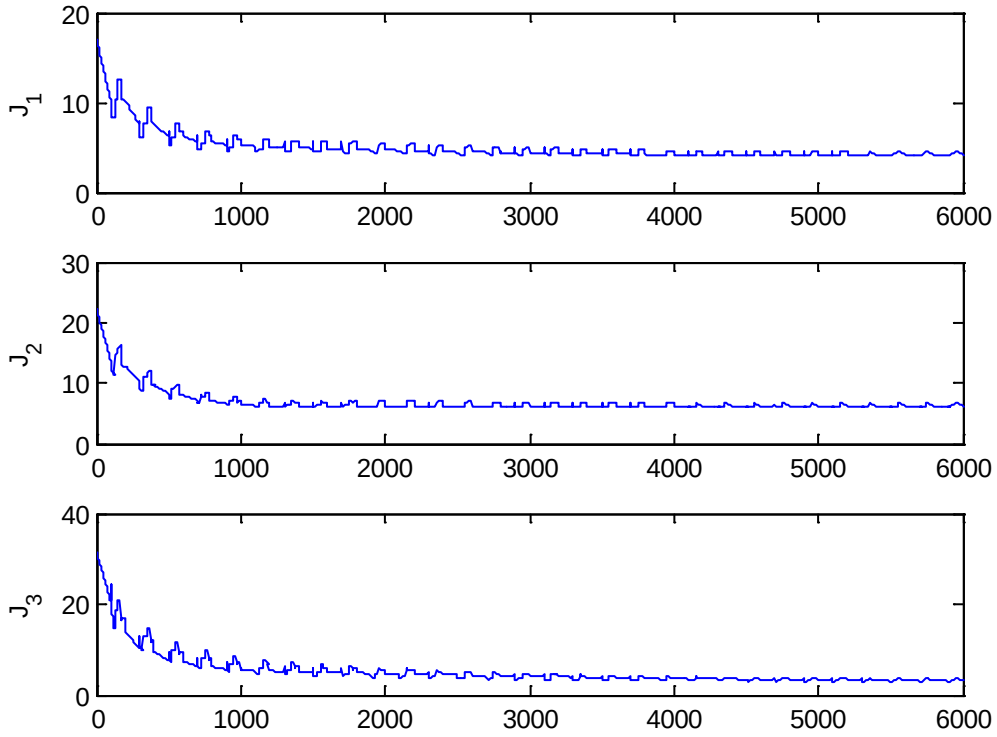

Fig 3-33: Output signals in multi-unit scheme with correction for three non-identical units (run 15)

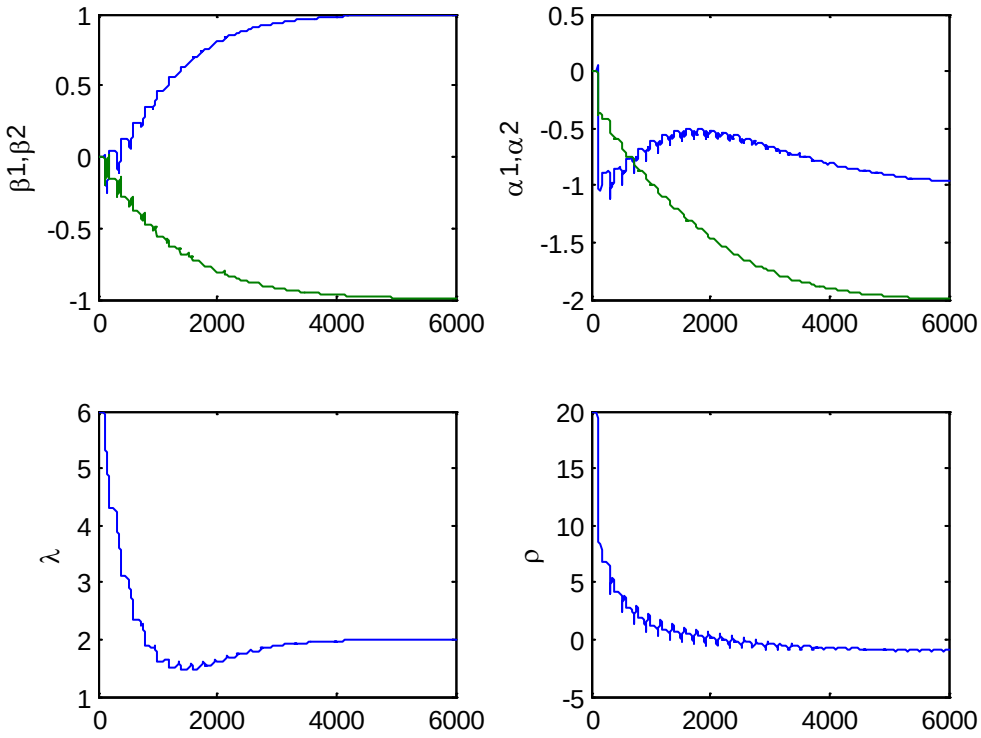

Fig 3-34: Estimated correctors in multi-unit scheme with correction for three non-identical units (run 15)

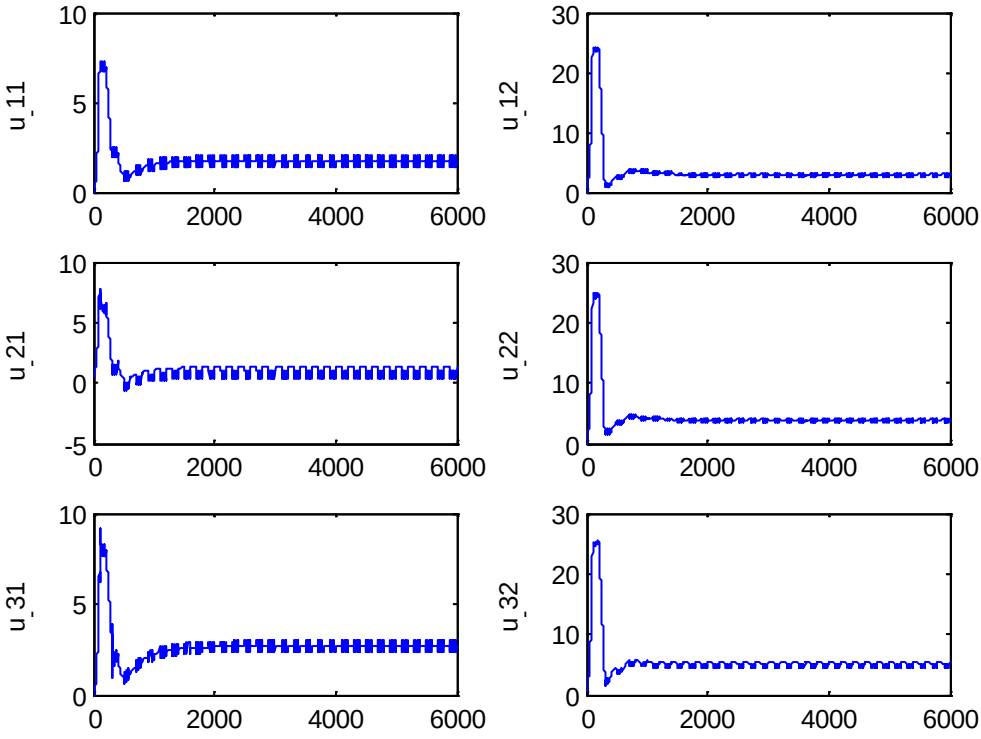

Fig 3-35: Input signals in multi-unit scheme with correction for three non-identical units (run 16)

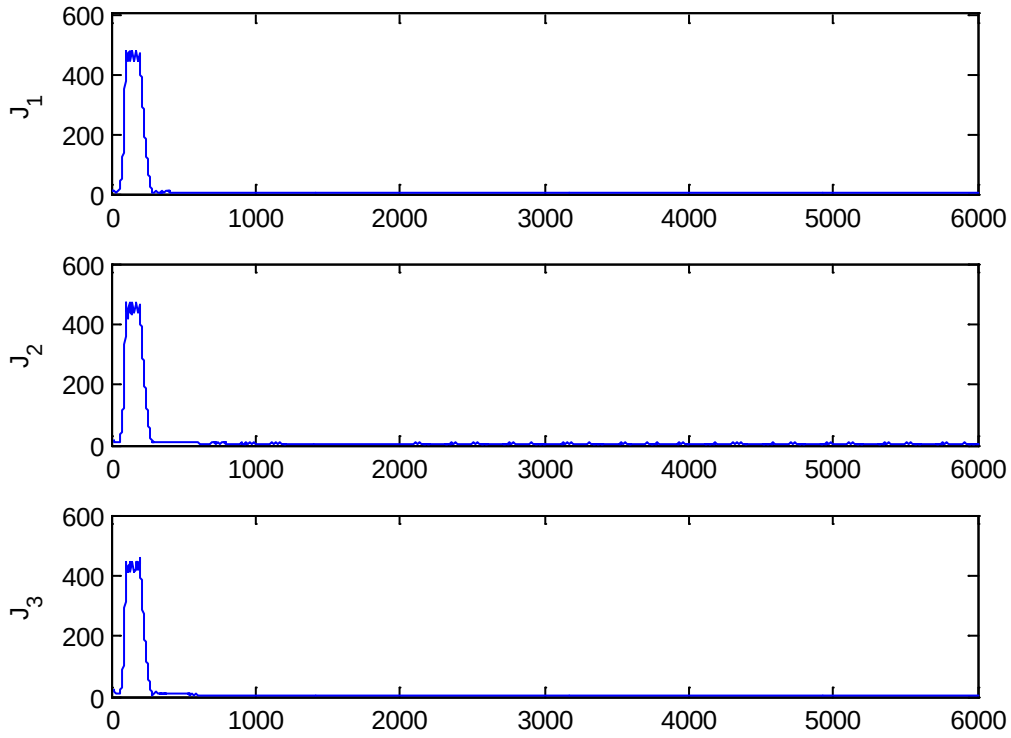

Fig 3-36: Output signals in multi-unit scheme with correction for three non-identical units (run 16)

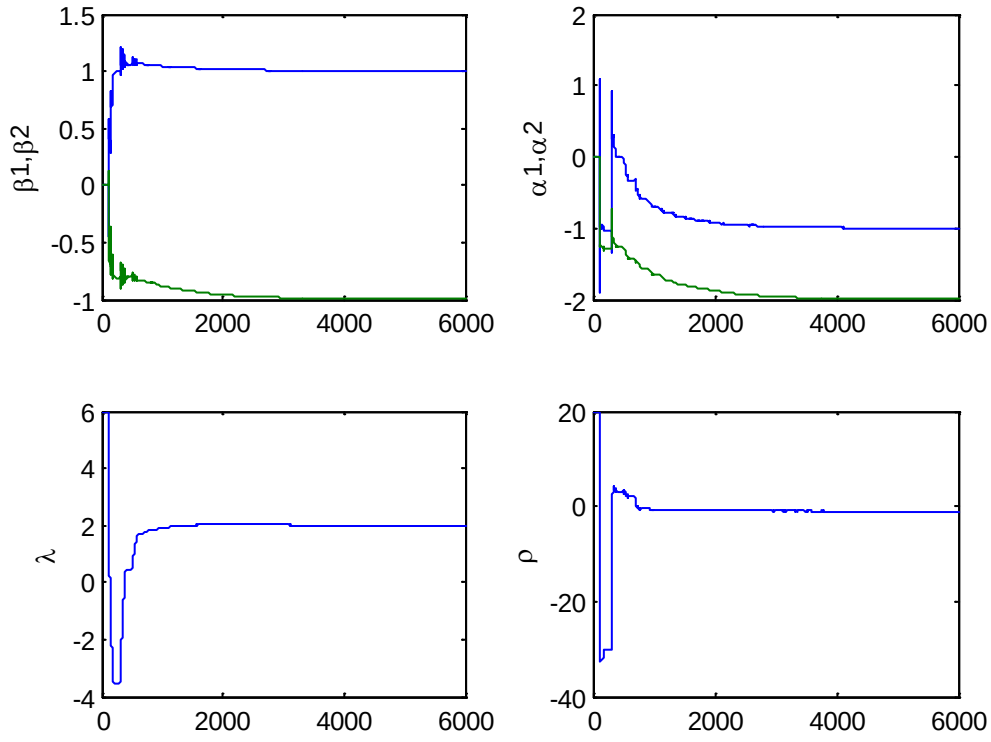

Fig 3-37: Estimated correctors in multi-unit scheme with correction for three non-identical units (run 16)

3.6 A guideline to tune parameters in multi-unit optimization algorithm

Based on the attempts made to choose parameters and gains in multi-unit algorithm, some heuristic rules are found. Although there is no certain method to tune these parameters, and the value of gains are derived based on trial and error, the following steps could be useful in running a multi-unit algorithm. It should be mentioned that the priori knowledge from the system is somehow needed in order to tune the parameters in multi-unit method.

First, based on the position of optimal points on the static curves, the offset Δ should be chosen. Next parameter is K_{mu} or the gain of integrator in multi-unit phase. After that, the parameter a which shows the amplitude of correction signal, and the periods T_1, and T_2 should be tuned. Then an initial point should be selected. Following this step, the signs for gains of correctors should be verified as it is presented in section 3.4.2. Finally the values for gains of correctors should be found by trial and error.

3.7 Brief Conclusion

As a brief conclusion of the chapter, multi-unit optimization algorithm has been modified for three non-identical units and two inputs by proposing two suitable correction signals for the correction phase in the multi-unit scheme. The differences between units have been characterized and the adaptation laws for the input and correctors have been proposed in such a way that the algorithm converges to the optimal point. Besides, tuning the parameters and choosing the sign and values for the gains of correctors have been investigated.

4 CHAPTER 4: CASE STUDY

This chapter is dedicated to the second objective that is to apply multi-unit method to maximize output power of a PV array model.

4.1 Application 1: Two units and one input

4.1.1 PV cell/array modeling

As discussed in the literature review, two main models for PV cell are single-diode model and double-diode model. Mono-crystalline PV cell has the best efficiency among all commercially available technology. Because the single-diode model is the best model fitted for Mono-crystalline PV cell and because of some limitations to develop expressions for the I-V curve parameters in two-diode model, single-diode model is selected for modeling the PV cell/array in this research. A schematic of the single-diode circuit model is shown in Fig 4-1.

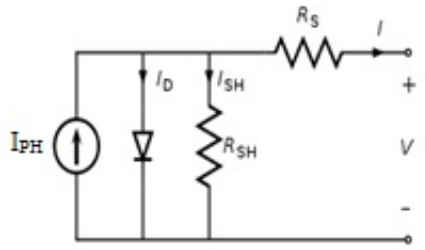

Fig 4-1: Single-diode model

According to Kirchhoff's current law:

$$\boldsymbol{I = I_{PH} - I_d - I_{SH}} \quad \text{Eq 46}$$

The voltage-current characteristic equation of a solar cell can be derived by following equations (Vachtsevanos and Kalaitzakis 1987).

Photo current: $\boldsymbol{I_{PH} = \left[I_{SC} + K_I\left(T_C - T_{Ref}\right)\right]\lambda_G}$ Eq 47

Diode current: $\boldsymbol{I_d = I_S\left[exp\left(\frac{q(V+IR_S)}{kT_CA}\right) - 1\right]}$ Eq 48

Cell's saturation current: $\boldsymbol{I_S = I_{RS}\left(\frac{T_C}{T_{Ref}}\right)^3 exp\left[\frac{qE_g}{kA}\left(\frac{1}{T_{Ref}} - \frac{1}{T_C}\right)\right]}$ Eq 49

Shunt current: $\boldsymbol{I_{SH} = \frac{V+IR_S}{R_{SH}}}$ Eq 50

$\lambda_G = \frac{G}{1000}$ Eq 51

In these equations I is output current (A), V is voltage across the output terminal (V), V_{OC} is PV open-circuit voltage, I_{SC} is short-circuit current at a 25°C and a insolation of 1kW/m^2, I_{RS} is the diode reverse saturation current, E_g is the band gap energy (eV), K_I is the cell's short-circuit current temperature coefficient (mA/°K), T_{Ref} and T_C are the cell's reference and current temperatures (°K), G is the insolation or the intensity of solar radiation (kW/m^2), A is diode ideality factor (between 1 and 1.5), q is elementary charge (1.6021×10^{-19}C), and k is Boltzmann's constant(1.3806×10^{-23}J/°K).

Since a typical PV cell produces less than 2W at 0.5V-0.8V (depending on the cell technology) approximately, the cells must be connected in series-parallel configuration to produce enough voltage and power (Tsai et al. 2008; Nema et al. 2009). A number of PV cells electrically connected to each other and mounted in a support structure or frame is called a PV module (panel). Multiple modules can be wired together to form an array. Fig 4-2 displays the position of PV cell in a PV module and the position of PV module in a PV array. In general, the larger the area of a module or array, the more electricity will be produced.

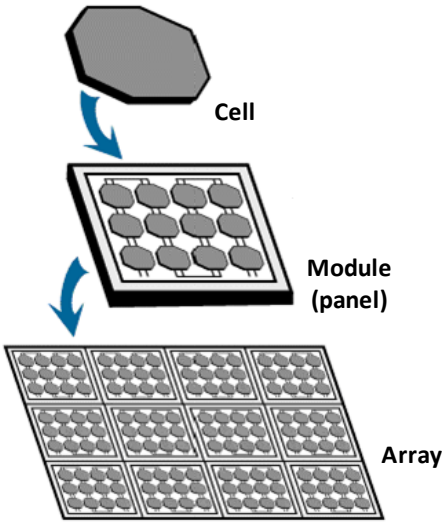

Fig 4-2: Schematic of PV cell in PV module and PV array (Knier 2002)

If we consider N_S cells in series to make a PV module, the terminal equation for the current and voltage (I-V) relationship for the PV module is given by:

$$I = I_{PH} - N_P I_S \left[exp\left(\frac{q\left(\frac{V}{N_S} + IR_S\right)}{kT_C A} \right) - 1 \right] - \frac{\left(\frac{V}{N_S} + IR_S\right)}{R_{SH}} \quad \text{Eq 52}$$

The terminal equation for the current and voltage of the array arranged in N_P parallel and N_S series becomes: (Tsai et al. 2008)

$$\boldsymbol{I = N_P I_{PH} - N_P I_S \left[exp\left(\frac{q\left(\frac{V}{N_S} + \frac{IR_S}{N_P}\right)}{kT_C A} \right) - 1 \right] - \frac{\left(\frac{N_P V}{N_S} + IR_S\right)}{R_{SH}}} \quad \text{Eq 53}$$

For the simulation in this chapter, the numerical values for Eq 46 to Eq 50 are picked from the manufacture's datasheet of the PV module 215N from Sanyo (Ghaffari et al. 2012). These values are presented in the Table 4-1.

Table 4-1: Numerical values from PV module 215N Sanyo

R_S	0.00248 [Ω]
R_{SH}	8.7 [Ω]
E_g	1.16 [eV]

A	1.81
K_I	1.96e-3 [mA/°K]
I_{SC}	5.61 [A]
I_{RS}	1.13e-6 [A]

A PV cell is modeled by MATLAB using a single diode model as a basic example. Current-voltage (I-V) and power-voltage (P-V) characteristics of this PV cell are shown in the following figures. In Fig 4-3, and Fig 4-4 it is assumed that $\lambda_G = 1$ and the curves are shown for different temperatures. It is observable that in fixed insolation, by increasing the temperature, short-circuit current of the PV cell is increased, whereas the maximum power point (MPP) is decreased. Therefore, the efficiency is decreased.

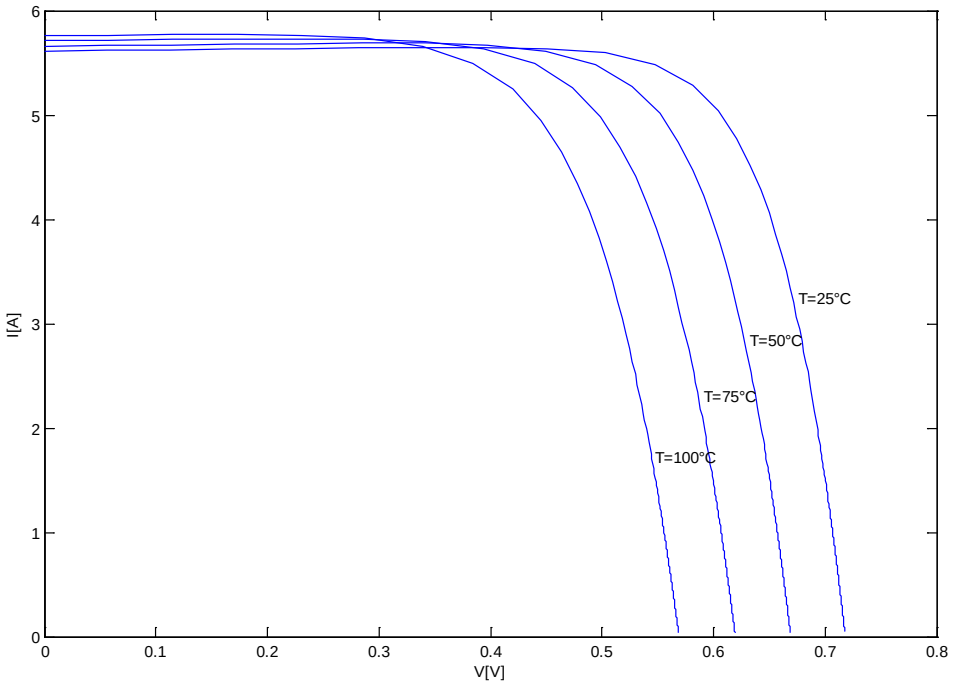

Fig 4-3: I-V characteristics of PV cell for different temperatures and $\boldsymbol{\lambda_G = 1}$

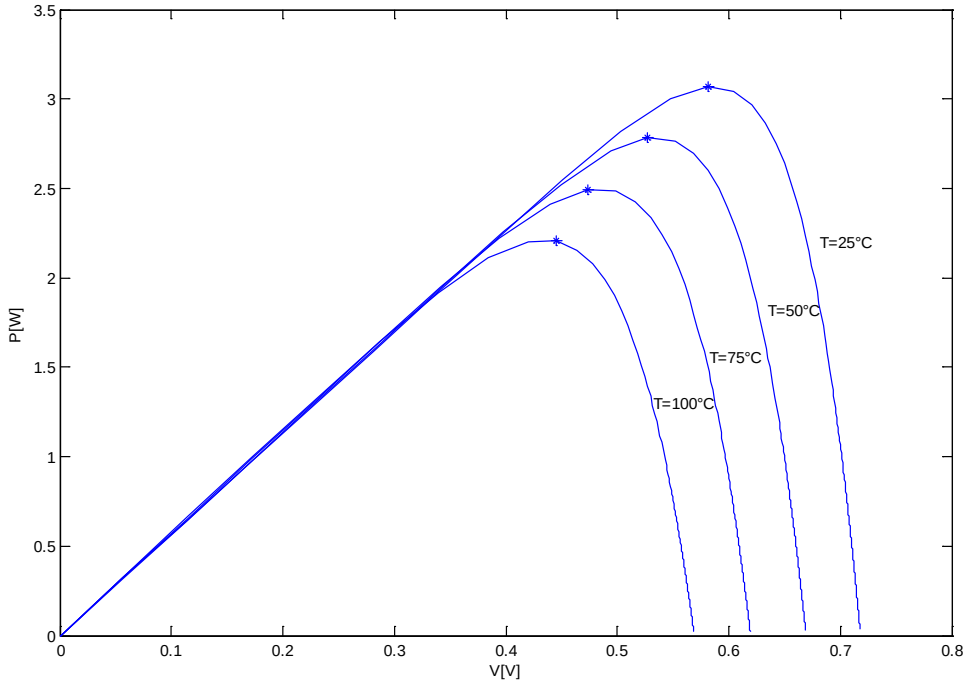

Fig 4-4: P-V characteristics of PV cell for different temperatures and $\boldsymbol{\lambda_G = 1}$

By simulating the model of PV cell, maximum power, optimal load or the resistance related to the MPP, and the voltage of MPP are found for different temperatures. In the Table 4-2, the results are presented.

Table 4-2: Optimal power, voltage, and load with $\boldsymbol{\lambda_G = 1}$

$T_C[°C]$	$\boldsymbol{P_{max}[W]}$	$\boldsymbol{R_{L-opt}[\Omega]}$	$\boldsymbol{V_{max}}\ [\boldsymbol{V}]$
25	3.0696	0.1100	0.5811
50	2.7825	0.1000	0.5275
75	2.4932	0.0900	0.4737
100	2.2052	0.0900	0.4455

In Fig 4-5 and Fig 4-6 it is assumed that $T_C = 25\,°C$ and the I-V and P-V curves are shown for different insolation. By increasing the insolation, the short-circuit current of the PV module is increased, and the MPP is increased as well.

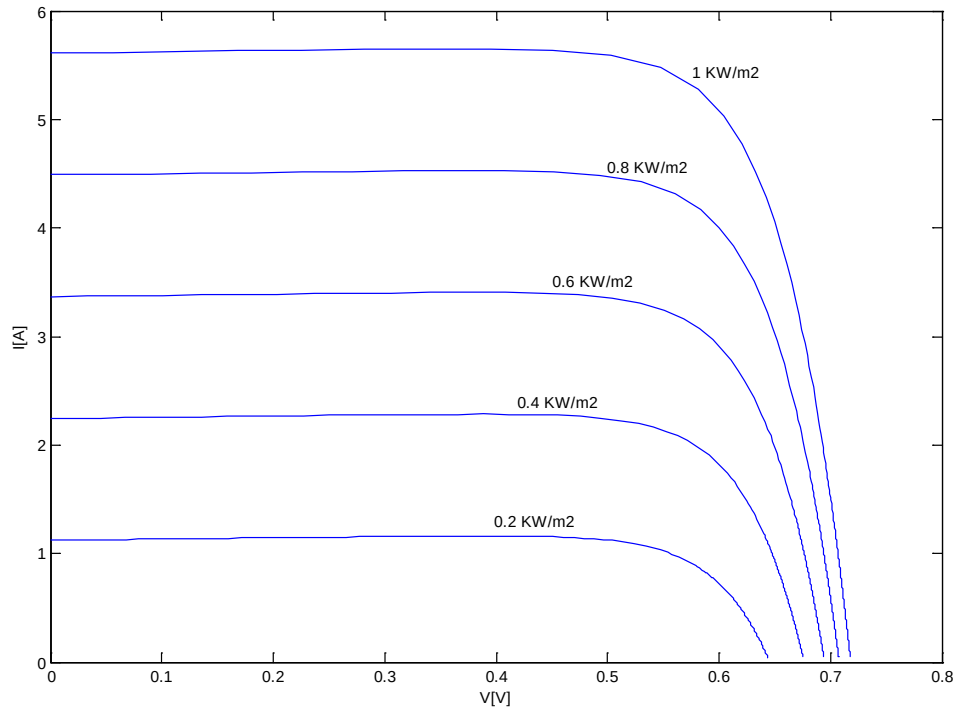

Fig 4-5: I-V characteristics of PV cell for different insolations and $\boldsymbol{T_C = 25\ °C}$

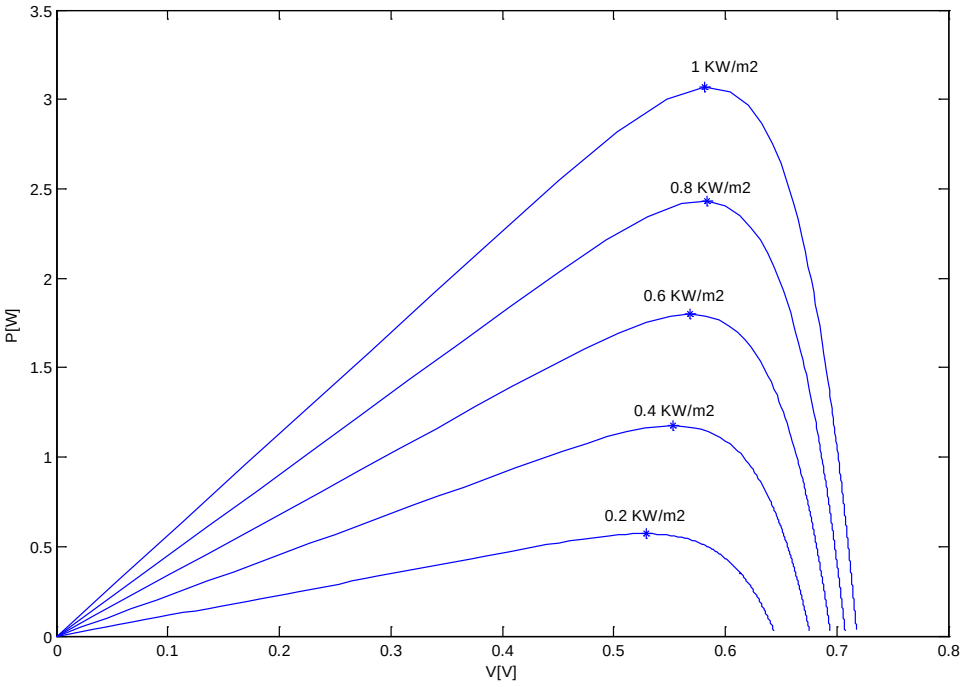

Fig 4-6: P-V characteristics of PV cell for different insolatios and $\boldsymbol{T_C = 25\ °C}$

In the Table 4-3, maximum power, optimal load or the resistance related to the MPP, and the voltage of MPP are shown for different insolations.

Table 4-3: Optimal power, voltage, and load with $\boldsymbol{T_C = 25\ °C}$

$\lambda_G[kW/m2]$	$P_{max}[W]$	$R_{L-opt}[\Omega]$	$V_{max}\ [V]$
0.2	0.5712	0.4900	0.5291
0.4	1.1753	0.2600	0.5528
0.6	1.7979	0.1800	0.5689
0.8	2.4296	0.1400	0.5832
1	3.0696	0.1100	0.5811

As it is noted, by connecting PV cells in series and parallel we can achieve more output power. For example by connecting N_S cells in series, P_{max} is multiplied by N_S (increasing). Moreover, by connecting N_P cells in parallel P_{max} is multiplied by N_P (increasing). In Table 4-4, maximum power, optimal load, and the voltage of MPP are shown for different configuration of PV cells. In these simulation $T_C = 25\ °C$ and $\lambda_G = 1$. As it is seen, when N_S=2 and N_P=2 the maximum power is equal to 12.0063 which is 4 times greater than the maximum power of N_S=1 and N_P=1. In the PV module 215N of Sanyo N_S=72 and N_P=1. We use this module for the optimization problem in this research.

Table 4-4: Optimal power, voltage, and load for different configuration of PV cells with $\boldsymbol{T_C = 25\ °C}$ and $\boldsymbol{\lambda_G = 1}$

N_S	N_P	$P_{max}[W]$	$R_{L-opt}[\Omega]$	$V_{max}\ [V]$
1	1	3.0016	0.1	0.5479
1	2	6.0032	0.2	1.0957
2	1	6.0032	0.2	1.0957
2	2	12.0063	0.1	1.0957
36	1	110.5450	4.02	21.0806
72	1	221.0905	8.03	42.135
72	2	442.1801	4.02	42.1612

4.1.2 Formulation the optimization problem

Based on Jacobi's law, a power source will deliver its maximum power to a load when the load has the same impedance as the internal impedance of the power source. But in general, real loads are far from the ideal load for a PV array and this mismatch results in major efficiency losses. Besides, the output current of PV arrays depend on atmospheric conditions such as temperature and insolation. These parameters are regularly changing so it's important to track the maximum power point to keep a maximum efficiency at

every instant. Based on these facts and the PV array model, optimization problem is formulated as followed:

$$\mathrm{Max}_R\ P_{PV}(R) \qquad \text{Eq 54}$$

In order to have a suitable problem for applying multi-unit method, in all simulations the optimization problem is translated to a minimization problem or $\mathrm{Min}_R(-P_{PV}(R))$. So the objective function is output power and the decision variable or the input of multi-unit algorithm is the load resistance. The disturbance inputs are ambient temperature (°K) and insolation (kW/m^2).

In Fig 4-7, the power curvature with respect to different loads (P-R curve) is shown for different temperature and $\lambda_G = 1$. This static curve is convex so we can apply multi-unit optimization method to obtain the maximum power.

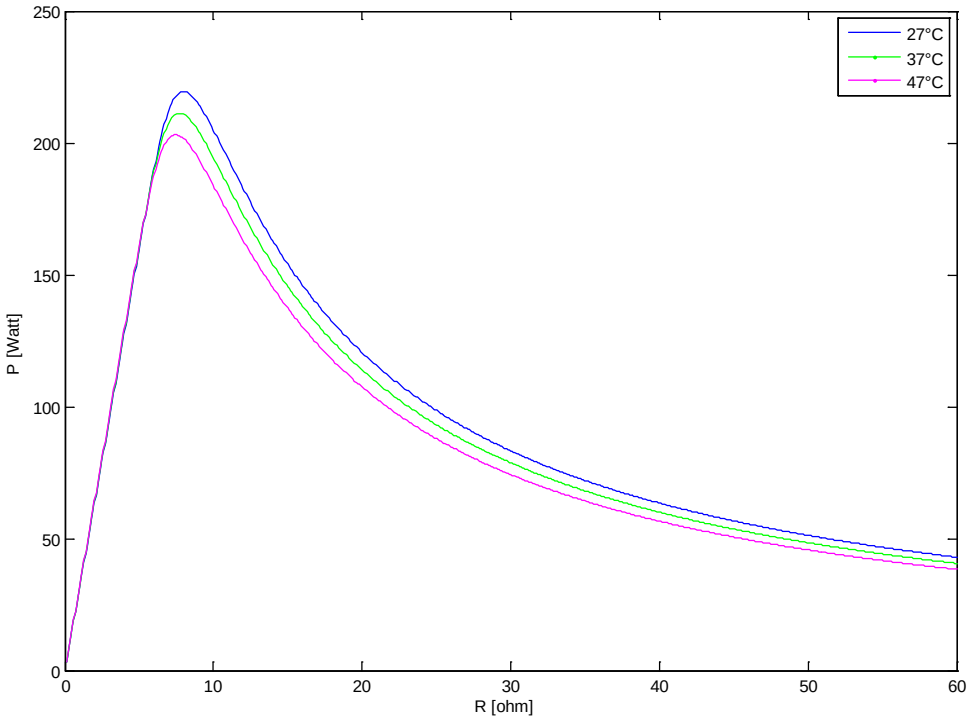

Fig 4-7: P-R curves for PV module 215N from Sanyo with N_S=72 and N_P=1

4.1.3 Multi-unit optimization for two identical PV arrays

The multi-unit algorithm is used to find the maximum power and optimal load related to that maximum value. In this part the case of two identical units is discussed. Therefore, it is assumed that the configuration of both arrays is the same, and also the conditions (temperature and insolation) in which they work are similar. Based on the Fig 4-7, the input of each unit is R and the output of each unit is P.

In Table 4-5 the results of applying multi-unit scheme on different configuration of the PV module 215N Sanyo with true optimal values extracted by MATLAB optimization toolbox are shown. In all runs $T_C = 25\ °C$ and $\lambda_G = 1$. By tuning the gain of integrator and selecting the offset Δ properly, the results achieved by multi-unit optimization method are the same as their real values extracted by "fminunc" or "fminsearch" functions in MATLAB. The letter N in the table represents the non-convergence multi-unit scheme.

Table 4-5: Optimal power and load for PV module 215N Sanyo with $\boldsymbol{T_C = 25\ °C}$, $\boldsymbol{\lambda_G = 1}$

N_S	N_P	Multi unit				fminunc or fminsearch	
		$P_{max}[W]$	$R_{L-opt}[\Omega]$	K	Δ	$P_{max}[W]$	$R_{L-opt}[\Omega]$
72	1	221.09	8.03	-100	0.3	221.09	8.03
72	2	N	N	-100	0.5	442.18	4.02
72	2	442.18	4.02	-1.5	0.2	442.18	4.02
72	3	663.27	2.68	-0.5	0.1	663.27	2.68

4.1.4 Multi-unit optimization for two non-identical PV arrays

In this part, the multi-unit algorithm including the correction phase is applied for two non-identical PV arrays. This algorithm works well for non-identical units when the shapes of the graphs are the same and they can be somehow fitted to each other by a slight shifting. To have different PV arrays we define three scenarios for configurations of PV module 215N Sanyo. The differences could be in their temperature, insolation, or number of parallel cells. Accordingly these are the defined scenarios:

1. Different T_C, same λ_G , and same N_P

2. Different $\boldsymbol{\lambda_G}$, same T_C, same N_P
3. Different N_P, same T_C, same $\boldsymbol{\lambda_G}$

Fig 4-8 shows the power-resistor (P-R) curve for a PV array contains one PV module 215N Sanyo with $\lambda_G = 1$ and different temperatures. This figure is related to the scenario 1. Fig 4-9 shows the PR curve for a PV array which contains one PV module 215N Sanyo with $T_C = 25\ °C$ and different insolations. This figure is related to the scenario 2. Finally, Fig 4-10 shows the P-R curve for a PV array contains different configuration of PV module 215N Sanyo with $T_C = 25\ °C$ and $\lambda_G = 1$. This means that the conditions are the same but the number of parallel cells is different which is related to the scenario 3.

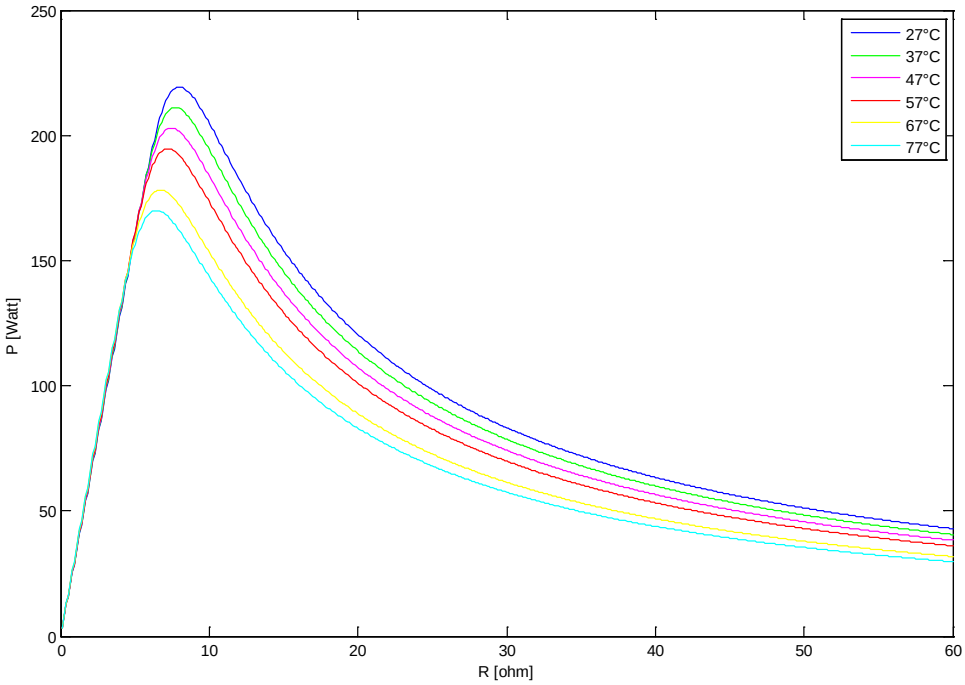

Fig 4-8: P-R Curves for a PV array with $\boldsymbol{\lambda_G = 1}$, $\boldsymbol{N_S = 72}$, $\boldsymbol{N_P = 1}$, and different temperatures

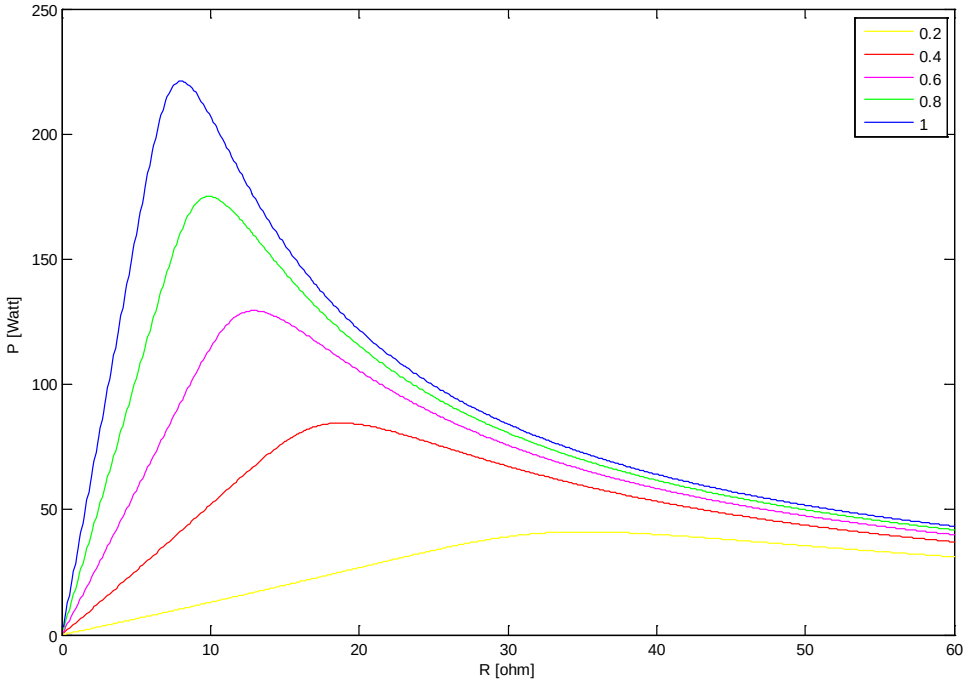

Fig 4-9: P-R Curves for a PV array with $\boldsymbol{T_C = 25\ °C}$, $\boldsymbol{N_S = 72}$, $\boldsymbol{N_P = 1}$, and different insolations

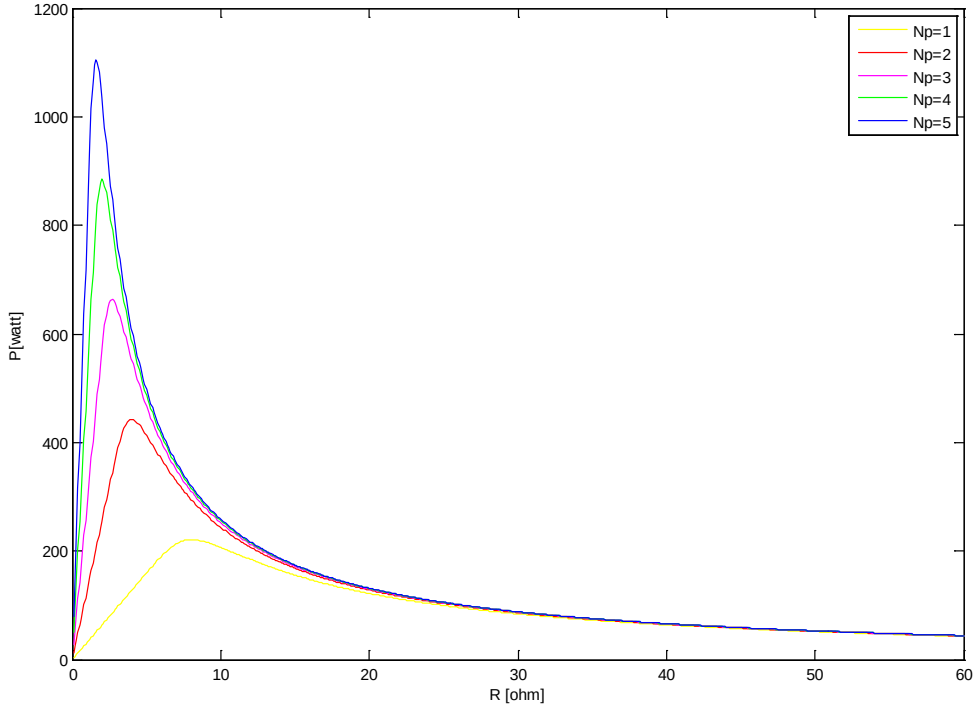

Fig 4-10: P-R Curves for a PV array with $\boldsymbol{T_C = 25\ °C}$, $\boldsymbol{\lambda_G = 1}$, $\boldsymbol{N_S = 72}$, and different numbers for parallel cells

Based on Fig 4-8, Fig 4-9, and Fig 4-10, and because of the similarities between the curves, two curves are chosen from each figure to form the two different units in final scenarios to apply multi-unit algorithm. Final scenarios are shown in the Table 4-6.

Table 4-6: Final scenarios to apply multi-unit algorithm for two non-identical units

Scenario	N_{S1}	N_{S2}	N_{P1}	N_{P2}	T_{C1}	T_{C2}	λ_{G1}	λ_{G2}
1	72	72	1	1	30°C	25°C	1	1
2	72	72	1	1	25°C	25°C	0.8	1
3	72	72	4	5	25°C	25°C	1	1

4.1.4.1 Scenario 1

In this part the initial point for multi-unit algorithm is chosen $R_0 = 7\Omega$, so based on the two P-R curves for scenario 1, the initial values for correctors are $\beta_0 = R_0 - R_0 = 0$, and $\lambda_0 \cong P_2(R_0) - P_1(R_0) = 2.3$. Among the numerous runs for this scenario, 8 runs are selected to show the importance of tuning the gains and parameters in multi-unit method with correctors. First, a proper value for Δ is chosen based on the difference between the optimal points on P-R static curves of each units. Then the parameter a is set. After that, the gains of multi-unit and correction phases are chosen with some trial and error. In Table 4-7 the values of parameters in each run are displayed. T_1 and T_2 are the periods for correction signals. Table 4-8 shows the optimal points and values, and the values of estimated correctors by applying the multi-unit with correctors for the 8 runs. The correct values of the optimal resistors and output powers for each unit which were calculated by MATLAB optimization toolbox are displayed in Table 4-9 to evaluate the functionality of multi-unit method. By comparing run 8 and 3, it can be realized that the ratio of Δ to a is crucially important to make the algorithm converges to optimal points.

Table 4-7: summary of parameters in applying multi-unit algorithm for two non-identical PV arrays (scenario 1)

Run	k_{mu}	Δ	a	k_β	k_λ	T_1	T_2
1	-0.01	0.5	0.5	0.001	0.01	200	100
2	-0.01	0.5	0.5	0.01	0.01	200	100
3	-0.01	-0.5	0.5	0.01	0.1	200	100
4	-0.1	0.5	0.5	0.01	0.01	200	100
5	-0.1	0.5	0.5	0.01	0.1	200	100
6	-0.1	0.5	0.2	0.01	0.1	200	50

7	-0.1	0.5	0.5	0.01	0.1	200	200
8	-0.1	0.5	0.09	0.01	0.1	200	100

Table 4-8: summary of results in applying multi-unit algorithm for two non-identical PV arrays (scenario 1)

Run	R_1^*	R_2^*	P_1^*	P_2^*	$\hat{\beta}^*$	$\hat{\lambda}^*$
1	7.4116	7.5429	215.3949	219.4715	-0.1312	4.0735
2	7.4287	7.6074	215.5098	219.8871	-0.1675	3.8224
3	7.4051	7.5365	215.3501	219.4273	-0.1314	4.0772
4	7.4287	7.6074	215.5098	219.8871	-0.1675	3.8224
5	7.4110	7.5423	215.3904	219.4675	-0.1313	4.0772
6	7.7079	7.8388	216.7577	220.8543	-0.1309	4.0966
7	7.6610	8.2923	216.6269	220.7040	-0.1313	4.0772
8	7.8174	7.9483	216.9479	221.0475	-0.1309	4.0996

Table 4-9: summary of results in using MATLAB optimization toolbox for each PV arrays (scenario 1)

R_1^*	R_2^*	P_1^*	P_2^*	β	λ
7.9	8.1	217	221.1	-0.2	4.1

Fig 4-11shows the graphs for input signals or resistors, and the corrector $\hat{\beta}$ for the run 8. The graphs for the corrector $\hat{\lambda}$, output power signals, and correction signal in run 8 are displayed in Fig 4-12.

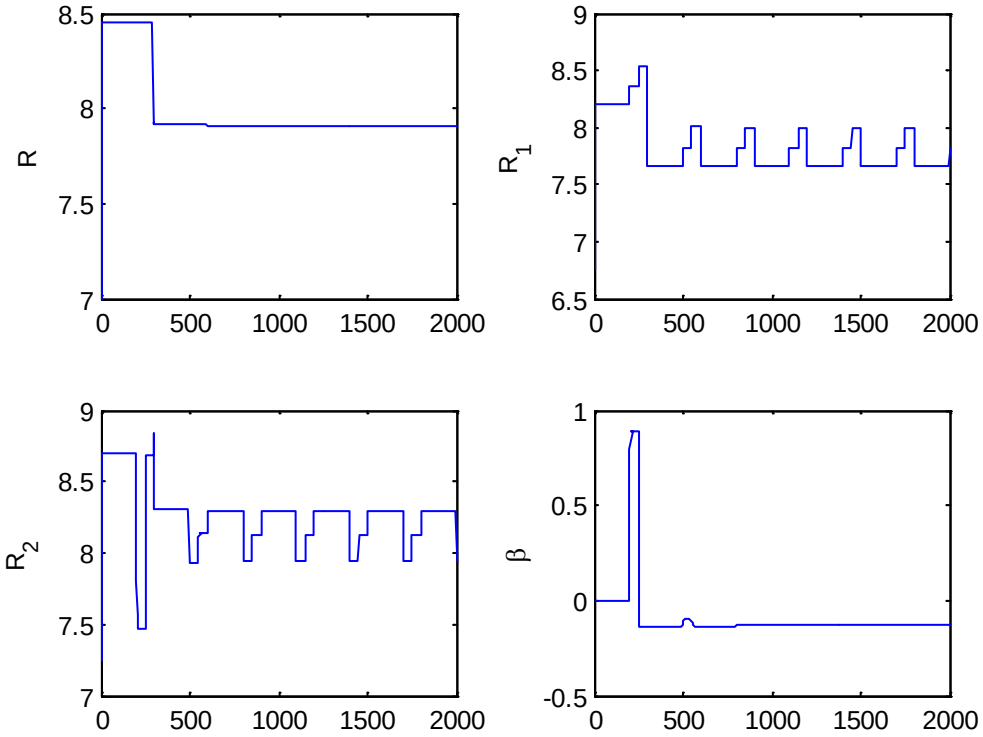

Fig 4-11: Resistors and corrector $\hat{\beta}$ in multi-unit scheme with correction for two non-identical PV arrays (run 8)

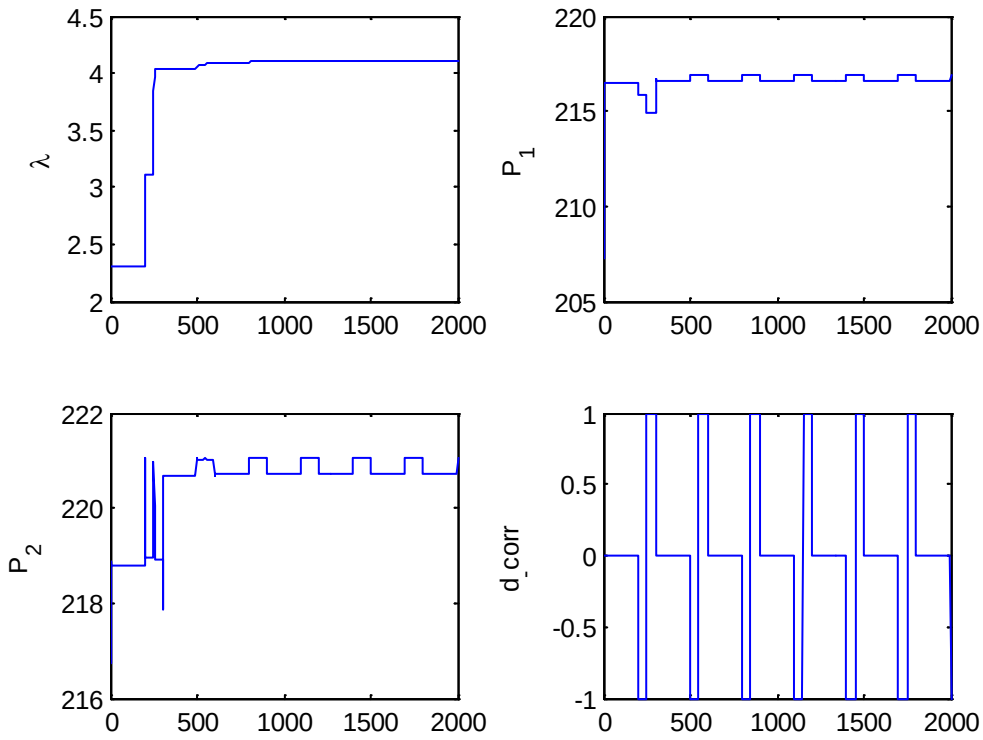

Fig 4-12: Output power signals, corrector $\hat{\boldsymbol{\lambda}}$, and correction signal in multi-unit scheme with correction for two non-identical PV arrays (run 8)

4.1.4.2 Scenario 2

In this part the initial point for multi-unit algorithm is chosen $R_0 = 7$, so based on the two P-R curves for scenario 2, the initial values for correctors are $\beta_0 = 0$, and $\lambda_0 = 0$. Among the numerous runs for this scenario, 2 runs are selected to show the importance of tuning the multi-unit gain. In Table 4-10 the values of parameters in each run are displayed. Table 4-11 shows the optimal points and values, and the values of estimated correctors by applying the multi-unit with correctors for the 2 runs. Finally, the correct values of the optimal resistors and output powers for each unit which were calculated by MATLAB optimization toolbox are displayed in Table 4-12. By comparing Table 4-11, and Table 4-12 it is realizable that in run 2 the optimal points and values are more near to their true values than in run 1.

Table 4-10: summary of parameters in applying multi-unit algorithm for two non-identical PV arrays (scenario 2)

Run	$\mathbf{k_{mu}}$	Δ	**a**	$\mathbf{k_\beta}$	$\mathbf{k_\lambda}$	$\mathbf{T_1}$	$\mathbf{T_2}$

1	-0.01	2	0.5	0.001	0.01	200	100
2	-0.1	2	0.5	0.01	0.1	200	100

Table 4-11: summary of results in applying multi-unit algorithm for two non-identical PV arrays (scenario 2)

Run	R_1^*	R_2^*	P_1^*	P_2^*	$\hat{\beta}^*$	$\hat{\lambda}^*$
1	9.0329	8.8670	171.0585	216.4053	1.8341	45.3468
2	9.3669	7.5329	174.0551	219.4016	1.834	45.3471

Table 4-12: summary of results in using MATLAB optimization toolbox for each PV arrays (scenario 2)

R_1^*	R_2^*	P_1^*	P_2^*	β	λ
10	8	175	221.1	2	46.1

Fig 4-13 shows the graphs for input signals or resistors, and the corrector $\hat{\beta}$ for the run 2. The graphs for the corrector $\hat{\lambda}$, output power signals, and correction signal in run 2 are displayed in Fig 4-14.

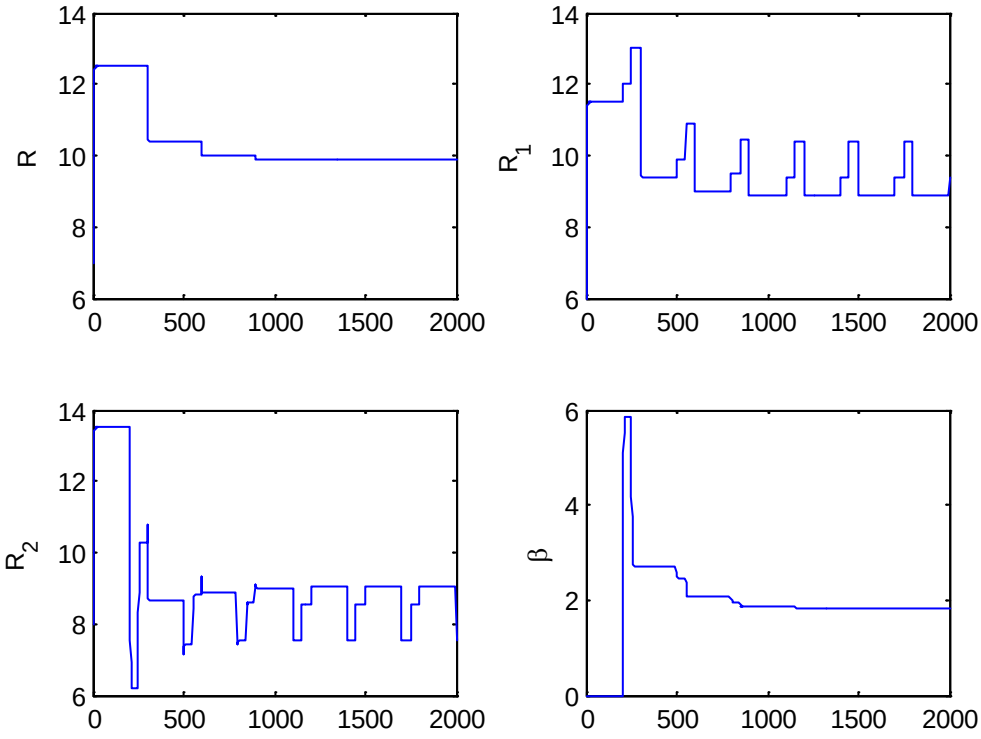

Fig 4-13: Resistors and corrector $\hat{\beta}$ in multi-unit scheme with correction for two non-identical PV arrays (run 2)

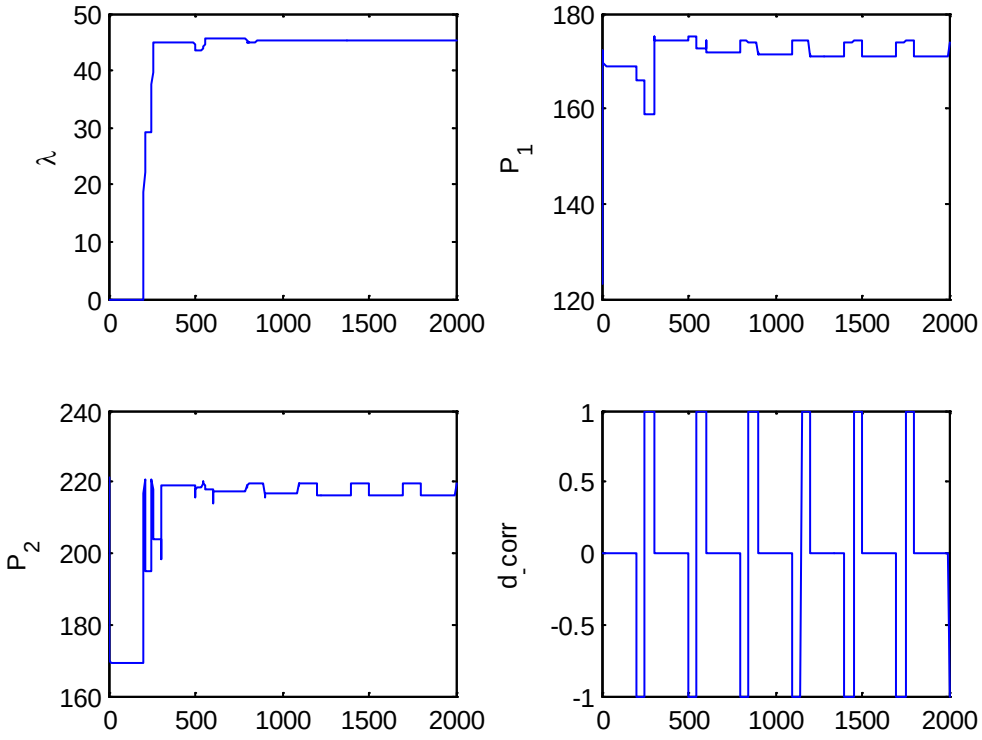

Fig 4-14: Output power signals, corrector $\hat{\boldsymbol{\lambda}}$, and correction signal in multi-unit scheme with correction for two non-identical PV arrays (run 2)

4.1.4.3 Scenario 3

In this part the initial point for multi-unit algorithm is chosen $R_0 = 1$, so based on the two P-R curves for scenario 3, the initial values for correctors are $\beta_0 = 0$, and $\lambda_0 = 200$. Among the numerous runs for this scenario, 6 runs are selected. In Table 4-13 the values of parameters in each run are displayed. Table 4-14 shows the optimal points and values, and the values of estimated correctors by applying the multi-unit with correctors for the 6 runs. The symbol N is chosen to show that the algorithm does not converge. The gains and parameters of the first two runs lead to convergence to optimal points and values. In run 3, changing the gain of multi-unit causes the algorithm to diverge. In run 4 the parameter a is decreasing comparing to the run 2 but the other parameters are the same. In run5 and run 6, the gains of correctors are changed which lead the algorithm not converging.

The correct values of the optimal resistors and output powers for each unit which were calculated by MATLAB optimization toolbox are displayed in Table 4-15.

Table 4-13: summary of parameters in applying multi-unit algorithm for two non-identical PV arrays (scenario 3)

Run	k_{mu}	Δ	a	k_{β}	k_{λ}	T_1	T_2
1	-0.001	1	0.2	0.0005	0.1	200	100
2	-0.001	0.4	0.08	0.0005	0.1	200	100
3	-0.01	0.4	0.08	0.0005	0.1	200	100
4	-0.001	0.4	0.05	0.0005	0.1	200	100
5	-0.001	0.4	0.08	0.0005	0.2	200	100
6	-0.001	0.4	0.08	0.001	0.2	200	100

Table 4-14: summary of results in applying multi-unit algorithm for two non-identical PV arrays (scenario 3)

Run	R_1^*	R_2^*	P_1^*	P_2^*	$\widehat{\beta}^*$	$\hat{\lambda}^*$
1	1.559	2.146	787	995	0.4126	208
2	1.796	1.805	864	1082	0.3912	219
3	N	N	N	N	N	N
4	N	N	N	N	N	N
5	N	N	N	N	N	N
6	N	N	N	N	N	N

Table 4-15: summary of results in using MATLAB optimization toolbox for each PV arrays (scenario 3)

R_1^*	R_2^*	P_1^*	P_2^*	β	λ
2	1.6	884.3	1105	0.4	220.7

Fig 4-15 and Fig 4-17 show the graphs for input signals or resistors, and the corrector $\hat{\beta}$ for the run 1 and run 2. The graphs for the corrector $\hat{\lambda}$, output power signals, and correction signal in run 1 and run 2 are displayed in Fig 4-16 and Fig 4-18 respectively. Comparing the results of these two runs shows that by choosing the smaller Δ and a in run 2, but keeping their ratio similar to run 1, the algorithm converges to more accurate optimal points and values. This ratio is deducted by several trial and errors.

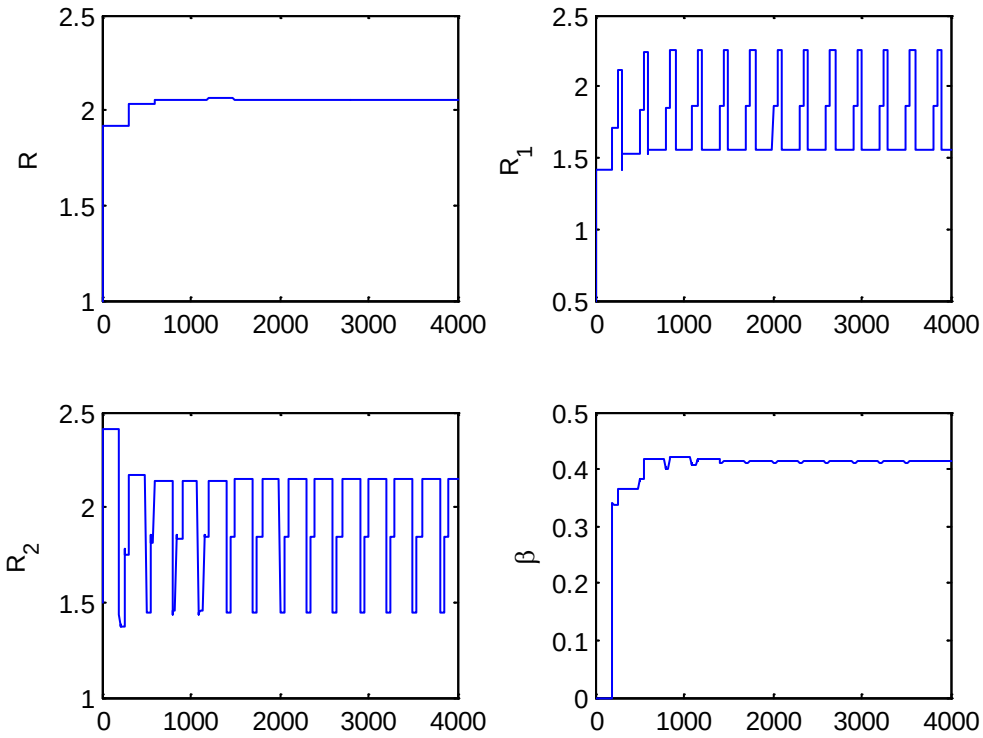

Fig 4-15: Resistors and corrector $\hat{\boldsymbol{\beta}}$ in multi-unit scheme with correction for two non-identical PV arrays (run 1)

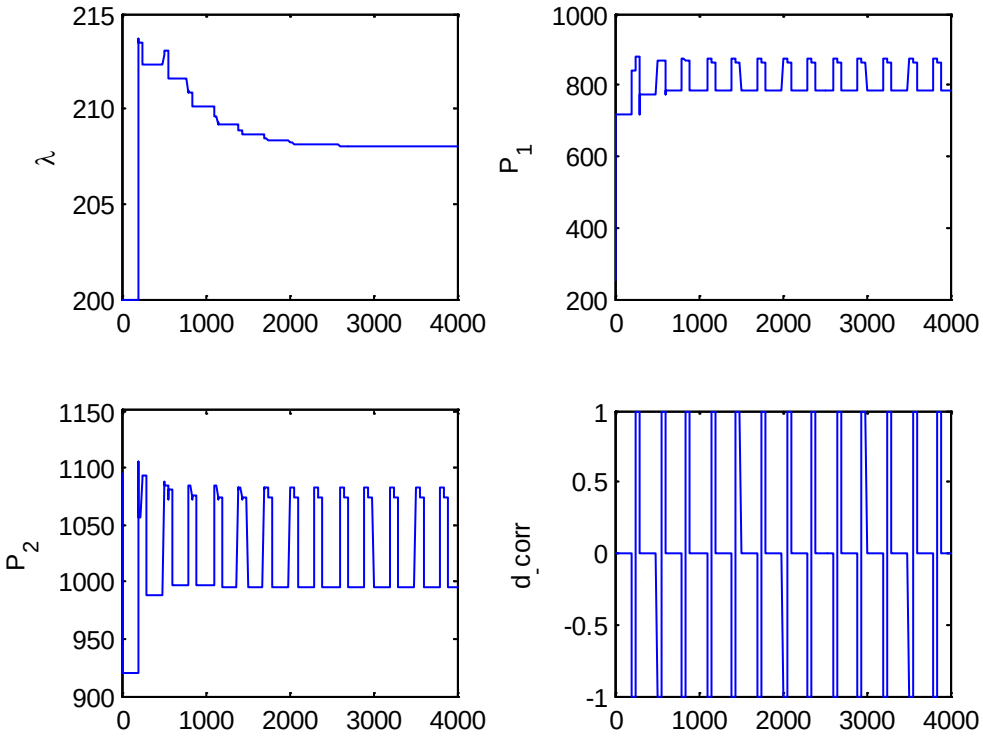

Fig 4-16: Output power signals, corrector $\hat{\boldsymbol{\lambda}}$, and correction signal in multi-unit scheme with correction for two non-identical PV arrays (run 1)

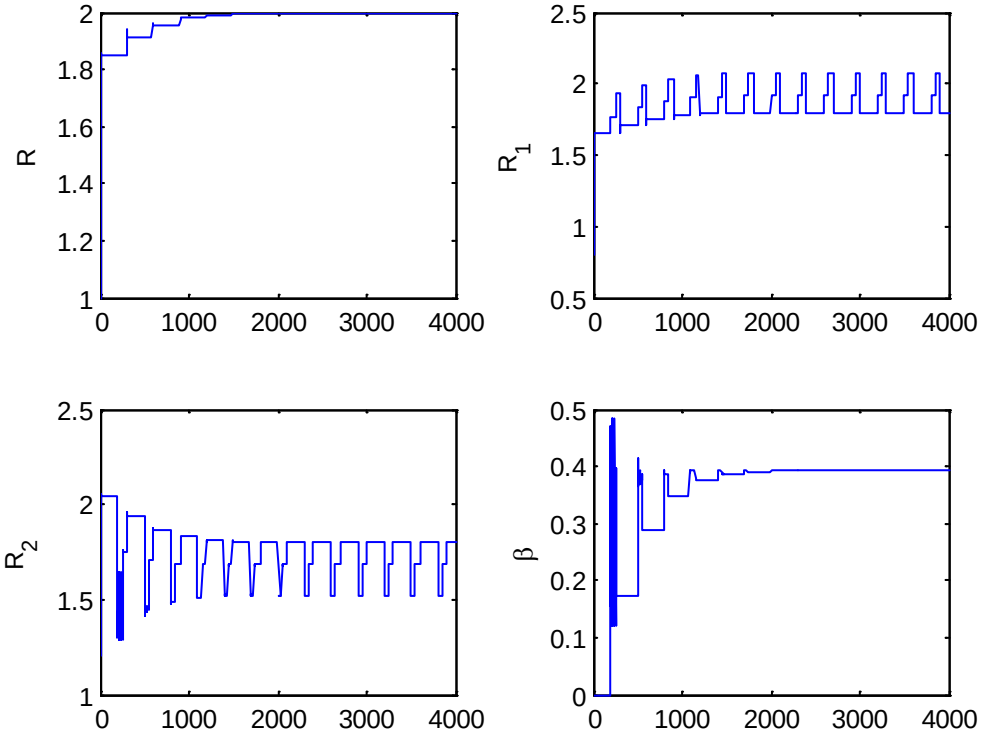

Fig 4-17: Resistors and corrector $\widehat{\boldsymbol{\beta}}$ in multi-unit scheme with correction for two non-identical PV arrays (run 2)

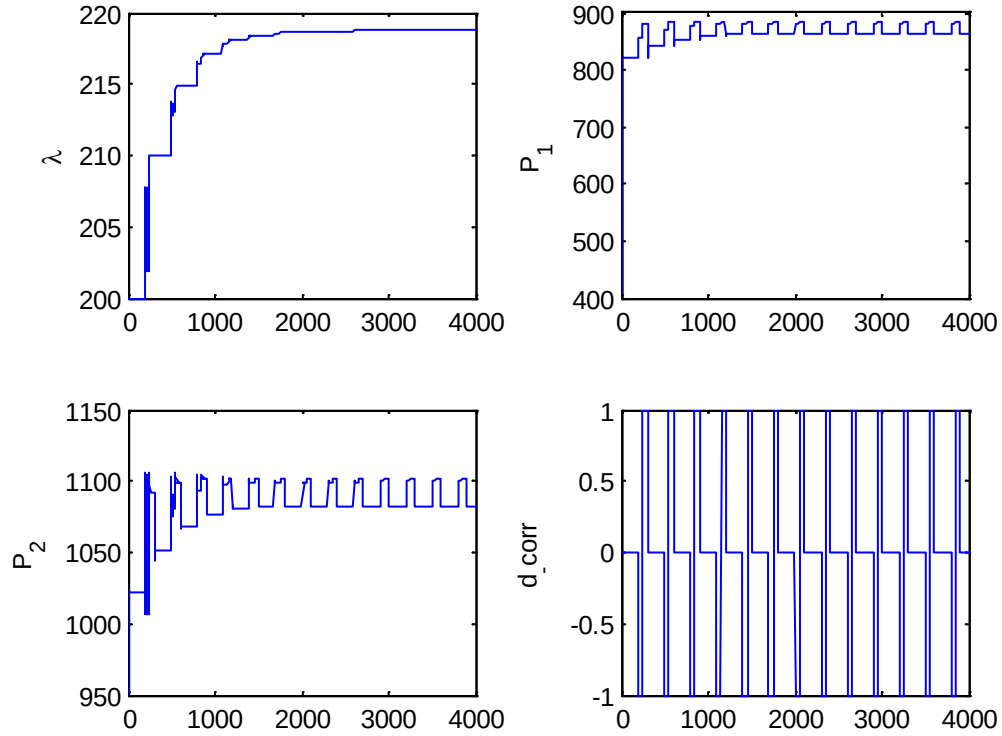

Fig 4-18: Output power signals, corrector $\hat{\boldsymbol{\lambda}}$, and correction signal in multi-unit scheme with correction for two non-identical PV arrays (run 2)

4.2 Brief conclusion

In this chapter, a single-diode model for a PV cell and a PV array has been presented. This model has been made and the IV and PV characteristics of the model have been simulated by MATLAB. Then the optimization problem regarding to MPPT has been introduced and the multi-unit algorithm has been applied on the model to solve the problem in both two identical and non-identical units.

5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

The main focus of this research is looking for possible ways to develop multi-unit optimization method with respect to the number of units and the number of inputs. There might be too many ways to think about this topic but we have tried to develop this algorithm for three non-identical units and two inputs to reach one step ahead of the previous works. Besides the theoretical aspect we tried to apply this method for PV arrays which can have many units in its nature. In particular, these are the achievements of this research:

- Propose an structure for multi unit optimization algorithm in the case of three non-identical units and two inputs
- Propose two different correction signals to compensate the differences between the units in the case of three non-identical units and two inputs
- Develop the adaptation laws needed for multi-unit phase and correction phase in the case of three non-identical units and two inputs

- Investigate some rule of thumb to tune gains for multi-unit phase and correction phase
- Find a proper model for different PV cell configuration in a manner that multi-unit method can be applied on them
- Apply the algorithm for different PV cell configuration in a PV array in the case of two units and one input
- Successfully find the maximum power point for PV arrays by the multi-unit algorithm

5.2 Recommendation

The following unexplored topics are recommended for future research:

- Stability analysis for the proposed algorithm in case of three non-identical units and two inputs
- Find a suitable real case study to apply the developed multi-unit algorithm in the case of three non-identical units and two inputs.

References

Adetola, V., D. DeHaan and M. Guay (2004). Adaptive extremum-seeking receding horizon control of nonlinear systems. American Control Conference, Boston, Massachusetts, IEEE.

Adetola, V. and M. Guay (2006). "Adaptive output feedback extremum seeking receding horizon control of linear systems." Journal of Process Control **16**(5): 521-533.

Adetola, V. and M. Guay (2007). "Parameter convergence in adaptive extremum-seeking control." Automatica **43**(1): 105-110.

Andersen, M. and B. Alvsten (1995). 200 W low cost module integrated utility interface for modular photovoltaic energy systems. IEEE 21st International Conference on Industrial Electronics, Control and Instrumentation, IEEE.

Ariyur, K. B. and M. Krstic (2003). Real-time optimization by extremum-seeking control, Wiley-Blackwell.

Astrom, K. J. and B. Wittenmark (1994). Adaptive control, Addison-Wesley Longman Publishing Co., Inc.

Azevedo, G., M. Cavalcanti, K. Oliveira, F. Neves and Z. Lins (2009). "Comparative evaluation of maximum power point tracking methods for photovoltaic systems." Journal of Solar Energy Engineering **131**: 031006.

Bamberger, W. and R. Isermann (1978). "Adaptive on-line steady-state optimization of slow dynamic processes." Automatica **14**(3): 223-230.

Banavar, R. (2003). "Extremum seeking loops with quadratic functions: estimation and control." International Journal of Control **76**(14): 1475-1482.

Becerra, V. M. and P. D. Roberts (1996). "Dynamic Integrated System Optimization and Parameter Estimation for Discrete Time Optimal Control fo Nonlinear Systems." International Jounral of Control **63**: 257–281.

Biegler, L. T., I. E. Grossmann and A. W. Westerberg (1985). "A note on approximation techniques used for process optimization." Computers & Chemical Engineering **9**(2): 201-206.

Bohórquez, M. A. M., J. M. Enrique Gómez and J. M. Andújar Márquez (2009). "A new and inexpensive temperature-measuring system: Application to photovoltaic solar facilities." Solar Energy **83**(6): 883-890.

Box, G. E. P. and N. R. Draper (1969). Evolutionary operation: a statistical method for process improvement, Wiley New York.

Brambilla, A., M. Gambarara, A. Garutti and F. Ronchi (1999). New approach to photovoltaic arrays maximum power point tracking. Power Electronics Specialists Conference, Charleston, SC , USA IEEE.

Bratcu, A. I., I. Munteanu, S. Bacha and B. Raison (2008). "Maximum power point tracking of grid-connected photovoltaic arrays by using extremum seeking control." Journal of Control Engineering and Applied Informatics **10**(4): 3-12.

Bright-Green-Energy. (2009). "Sizing a Solar Photovolatic (PV) Array and System." 13 October 2011, from http://www.wirefreedirect.com/solar_panel_sizing.asp.

Cabal, C., C. Alonso, A. Cid-Pastor, B. Estibals, L. Seguier, R. Leyva, G. Schweitz and J. Alzieu (2007). Adaptive digital MPPT control for photovoltaic applications. IEEE International Symposium on Industrial Electronics, Vigo, Spain IEEE.

Chachuat, B., A. Marchetti and D. Bonvin (2008). "Process optimization via constraints adaptation." Journal of Process Control **18**(3): 244-257.

Chachuat, B., B. Srinivasan and D. Bonvin (2009). "Adaptation strategies for real-time optimization." Computers & Chemical Engineering **33**(10): 1557-1567.

Chen, C. Y. and B. Joseph (1987). "On-line optimization using a two-phase approach: an application study." Industrial & engineering chemistry research **26**(9): 1924-1930.

Chioua, M., B. Srinivasan, M. Guay and M. Perrier (2007). "Dependence of the Error in the Optimal Solution of Perturbation-Based Extremum Seeking Methods on the Excitation Frequency." The Canadian Journal of Chemical Engineering **85**(4): 447-453.

Chiu, H. J., Y. K. Lo, T. P. Lee, Q. S. Chen, W. L. Yu, J. X. Lee, F. Shih and S. C. Mou (2011). "A battery charger with maximum power point tracking function for low-power photovoltaic system applications." International Journal of Circuit Theory and Applications **39**(3): 241-256.

Choi, J. Y., M. Krstic, K. B. Ariyur and J. S. Lee (2002). "Extremum seeking control for discrete-time systems." Automatic Control, IEEE Transactions on **47**(2): 318-323.

Darby, M. L. and D. C. White (1988). "On-line optimization of complex processes. Chemical Engineering Programme, , 51– 59." Chemical Engineering Programme(10): 51– 59.

DeHaan, D. and M. Guay (2004). Extremum seeking control of nonlinear systems with parametric uncertainties and state constraints. American Control Conference, Boston, Massachusetts, IEEE.

DeHaan, D. and M. Guay (2005). "Extremum-seeking control of state-constrained nonlinear systems." Automatica **41**(9): 1567-1574.

Drakunov, S., U. Ozguner, P. Dix and B. Ashrafi (1995). "ABS control using optimum search via sliding modes." Control Systems Technology, IEEE Transactions on **3**(1): 79-85.

Durbeck, R. (1965). Principles for simplification of optimizing control models, PhD thesis. PhD, Case Institute of Technology.

Electropaedia. (2005). "Solar Power." Retrieved 8 August 2011, from http://www.mpoweruk.com/solar_power.htm.

Enrique, J., J. Andújar and M. Bohórquez (2010). "A reliable, fast and low cost maximum power point tracker for photovoltaic applications." Solar Energy **84**(1): 79-89.

Findeisen, W., F. N. Bailey, M. Brdys, K. Malinowski, P. Tatjewski and A. Wozniak (1980). Control and coordination in hierarchical systems, J. Wiley Chichester-New York.

Forbes, J., T. Marlin and J. MacGregor (1994). "Model adequacy requirements for optimizing plant operations." Computers & Chemical Engineering **18**(6): 497-510.

Forbes, J. F. and T. E. Marlin (1994). "Model accuracy for economic optimizing controllers: the bias update case." Industrial & engineering chemistry research **33**(8): 1919-1929.

François, G., B. Srinivasan and D. Bonvin (2005). "Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty." Journal of Process Control **15**(6): 701-712.

Gao, W. and S. Engell (2005). "Iterative set-point optimization of batch chromatography." Computers & Chemical Engineering **29**(6): 1401-1409.

Garcia, C. E. and M. Morari (1981). "Optimal operation of integrated processing systems. Part I: Open-loop on-line optimizing control." AIChE Journal **27**(6): 960-968.

Ghaffari, A., S. Seshagiri and M. Krstic (2012). Power optimization for photovoltaic micro-converters using multivariable gradient-based extremum-seeking. American Control Conference (ACC), 2012.

Guay, M. and T. Zhang (2003). "Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainties." Automatica **39**(7): 1283-1293.

Hohm, D. P. and M. E. Ropp (2003) "Comparative study of maximum power point tracking algorithms." Progress in Photovoltaics: Research and Applications **11**, 47–62 DOI: 10.1002/pip.45.

Hua, C., J. Lin and C. Shen (2002). "Implementation of a DSP-controlled photovoltaic system with peak power tracking." Industrial Electronics, IEEE Transactions on **45**(1): 99-107.

Hussein, K., I. Muta, T. Hoshino and M. Osakada (1995). "Maximum photovoltaic power tracking: an algorithm for rapidly changing atmospheric conditions." IEE Proceedings-Generation, Transmission and Distribution **142**(1): 59-64.

Ioannou, P. A. and J. Sun (1996). Robust adaptive control, Prentice-Hall.

Joyce, A., C. Rodrigues and R. Manso (2001). "Modelling a PV system." Renewable energy **22**(1): 275-280.

Khalil, H. K. (2002). "Nonlinear systems." Prentice Hall, Inc.

Knier, G. (2002). "How do Photovoltaics Work?" Retrieved 4 Feb, 2013, from http://science.nasa.gov/science-news/science-at-nasa/2002/solarcells/.

Krauter, S. (2006). Introduction. Solar electric power generation-photovoltaic energy systems. Berlin, Springer.

Krstic, M. (2000). "Performance improvement and limitations in extremum seeking control." Systems & control letters **39**(5): 313-326.

Krstic, M., I. Kanellakopoulos and P. Kokotovic (1995). Nonlinear and adaptive control design, Wiley New York.

Krstic, M. and H. Wang (2000). "Stability of extremum seeking feedback for general nonlinear dynamic systems." AUTOMATICA **36**: 595-601.

Krstic, M., H. H. Wang and G. Bastin (1999). "Optimizing bioreactors by extremum seeking." International Journal of Adaptive Control and Signal Processing **13**(8).

Landau, Y. D. (1979). Adaptive control: The model reference approach. New York, Marcel Dekker.

Leblanc, M. (1922). "Sur l'electrification des chemins de fer au moyen de courants alternatifs de frequence elevee." Revue Generale de l'Electricite.

Leyva, R., C. Alonso, I. Queinnec, A. Cid-Pastor, D. Lagrange and L. Martinez-Salamero (2006). "MPPT of photovoltaic systems using extremum-seeking control." Aerospace and Electronic Systems, IEEE Transactions on **42**(1): 249-258.

Manzie, C. and M. Krstic (2009). "Extremum seeking with stochastic perturbations." Automatic Control, IEEE Transactions on **54**(3): 580-585.

Marchetti, A., B. Chachuat and D. Bonvin (2009). "Modifier-adaptation methodology for real-time optimization." Industrial & engineering chemistry research **48**(13): 6022-6033.

Marlin, T. E. and A. N. Hrymak (1997). Real-time operations optimization of continuous processes. Fifth International Conference on Chemical Process Control of AIChE Symposium Series, Lake Tahoe New York, NY: American Institute of Chemical Engineers, 1971-c2002.

McFarlane, R. and D. Bacon (1989). "Empirical strategies for open-loop on-line optimization." The Canadian Journal of Chemical Engineering **67**(4): 665-677.

Morari, M., Y. Arkun and G. Stephanopoulos (1980). "Studies in the synthesis of control structures for chemical processes: Part I: Formulation of the problem. Process decomposition and the classification of the control tasks. Analysis of the optimizing control structures." AIChE Journal **26**(2): 220-232.

Nema, R., S. Nema and G. Agnihotri (2009). "Computer simulation based study of photovoltaic cells/modules and their experimental verification." International Journal of Recent Trends in Engineering **1**(3): 151-156.

Nema, S., R. Nema and G. Agnihotri (2010). "Matlab/simulink based study of photovoltaic cells/modules/array and their experimental verification." International Journal of Energy and Environment: 487-500.

Nguang, S. and X. Chen (2000). "Extremum seeking scheme for continuous fermentation processes described by an unstructured fermentation model." Bioprocess and Biosystems Engineering **23**(5): 417-420.

Perkins, J. D. (1998). Plant-wide Optimization: Opportunities and Challenges. FACOPO, New York, New York, NY: American Institute of Chemical Engineers, 1971-c2002.

Petreuş, D., C. Fărcaş and I. Ciocan (2008). "Modelling And Simulation Of Photovoltaic Cells." Acta Napocensis, Electronics and Telecommunications: 42-47.

Pfaff, G., J. Fraser Forbes and P. James McLellan (2006). "Generating information for real-time optimization." Asia-Pacific Journal of Chemical Engineering **1**(1-2): 32-43.

Popovic, D., M. Jankovic, S. Magner and A. R. Teel (2006). "Extremum seeking methods for optimization of variable cam timing engine operation." Control Systems Technology, IEEE Transactions on **14**(3): 398-407.

Roberts, P. (1979). "An algorithm for steady-state system optimization and parameter estimation." International Journal of Systems Science **10**(7): 719-734.

Salameh, Z. M., F. Dagher and W. A. Lynch (1991). "Step-down maximum power point tracker for photovoltaic systems." Solar Energy **46**(5): 279-282.

Sequeira, S. E., M. Graells and L. Puigjaner (2002). "Real-time evolution for on-line optimization of continuous processes." Industrial & engineering chemistry research **41**(7): 1815-1825.

Sequeira, S. E., M. Herrera, M. Graells and L. Puigjaner (2004). "On-line process optimization: parameter tuning for the real time evolution (RTE) approach." Computers & Chemical Engineering **28**(5): 661-672.

Skogestad, S. (2000). "Plantwide control: The search for the self-optimizing control structure." Journal of Process Control **10**(5): 487-507.

Skogestad, S. (2000). "Self-optimizing control: The missing link between steady-state optimization and control." Computers & Chemical Engineering **24**(2-7): 569-575.

Srinivasan, B. (2007). "Real time optimization of dynamic systems using multiple units." International Journal of Robust and Nonlinear Control **17**(13): 1183-1193.

Srinivasan, B. and D. Bonvin (2007). "Real-time optimization of batch processes by tracking the necessary conditions of optimality." Industrial & engineering chemistry research **46**(2): 492-504.

Sternby, J. (1980). Extremum control systems- An area for adaptive control. Joint American Control Conference, San Fancisco, CA.

Tsai, H. L., C. S. Tu and Y. J. Su (2008). Development of generalized photovoltaic model using MATLAB/SIMULINK. Proceedings of the World Congress on Engineering and Computer Science San Francisco, USA, Citeseer.

Vachtsevanos, G. and K. Kalaitzakis (1987). "A Hybrid Photovoltaic Simulator for Utility Interactive Studies." Energy Conversion, IEEE Transactions on **EC-2**(2): 227-231.

Villalva, M. G. and J. R. Gazoli (2009). "Comprehensive approach to modeling and simulation of photovoltaic arrays." Power Electronics, IEEE Transactions on **24**(5): 1198-1208.

Wang, H. H., S. Yeung and M. Krstic (2000). "Experimental application of extremum seeking on an axial-flow compressor." Control Systems Technology, IEEE Transactions on **8**(2): 300-309.

Wasynezuk, O. (2007). "Dynamic behavior of a class of photovoltaic power systems." Power Apparatus and Systems, IEEE Transactions on(9): 3031-3037.

White, D. (1997). "Online optimization: what, where and estimating ROI: Process optimization: Process control and instrumentation." Hydrocarbon Processing **76**(6): 43-51.

Won, C. Y., D. H. Kim, S. C. Kim, W. S. Kim and H. S. Kim (1994). A new maximum power point tracker of photovoltaic arrays using fuzzy controller. 24th IEEE Power Electronics Specialists Conference, IEEE.

Woodward, L. (2009). Adaptation de la Méthode Multi-Unités à l'Optimisation sous Contraintes en Présence d'Unités Non Identiques, Ph.D. Thesis. DU DIPLÔME DE PHILOSOPHIÆ DOCTOR, Ecole Polytechnique de Montreal

Woodward, L., M. Perrier and B. Srinivasan (2007). Convergence of multi-unit optimization with non-identical units: Application to the optimization of a bioreactor. 7th IFAC Symposium on Nonlinear Control Systems Pretoria, South Africa.

Woodward, L., M. Perrier and B. Srinivasan (2007). Multi-unit optimization with gradient projection on active constraints. 8th International IFAC Symposium on Dynamics and Control of Process Systems, Cancún, Mexico.

Woodward, L., M. Perrier and B. Srinivasan (2009a). "Improved performance in the multi-unit optimization method with non-identical units." Journal of Process Control **19**(2): 205-215.

Woodward, L., M. Perrier and B. Srinivasan (2010). A Simultaneous Approach for Correcting Differences between Units in Multi-unit Optimization. Proceedings of the 9th International Symposium on Dynamics and Control of Process Systems Leuven, Belgium.

Woodward, L., B. Tartakovsky, M. Perrier and B. Srinivasan (2009b). "Maximizing power production in a stack of microbial fuel cells using multiunit optimization method." Biotechnology Progress **25**(3): 676-682.

Zhang, C. and R. Ordónez (2005). Numerical optimization-based extremum seeking control of LTI systems. 44th IEEE Conference on Decision and Control, and the European Control Conference, IEEE.

Zhang, C. and R. Ordónez (2006). Extremum Seeking Control based on Numerical Optimization and State Regulation-Part I: Theory and Framework. 45th IEEE Decision and Control, IEEE.

Zhang, H. and P. Roberts (1991). "Integrated system optimization and parameter estimation using a general form of steady-state model." International Journal of Systems Science **22**(10): 1679-1693.

Zhang, Y. and J. F. Forbes (2000). "Extended design cost: a performance criterion for real-time optimization systems." Computers & Chemical Engineering **24**(8): 1829-1841.

Zhang, Y. and J. F. Forbes (2006). "Performance Analysis of Perturbation-Based Methods for Real Time Optimization." The Canadian Journal of Chemical Engineering **84**(2): 209-218.

Zhang, Y., D. Monder and J. Fraser Forbes (2002). "Real-time optimization under parametric uncertainty: a probability constrained approach." Journal of Process Control **12**(3): 373-389.